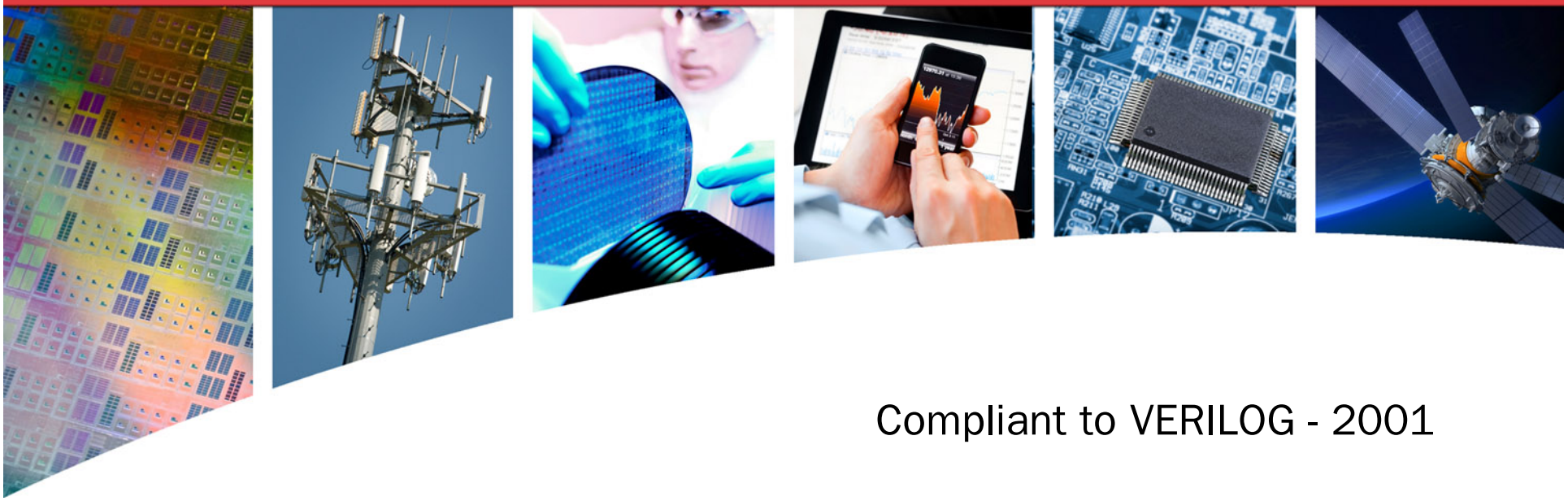
Silos – Verilog Simulator



Compliant to VERILOG - 2001

SILVACO

Agenda

- Introduction
- Starting Silos Project
- Explorer and Analyzer
- Source Code Debugging
- State Machine Design Entry
- Advanced Debugging Features
- Finite State Machine Example
- Gate Level Debugging

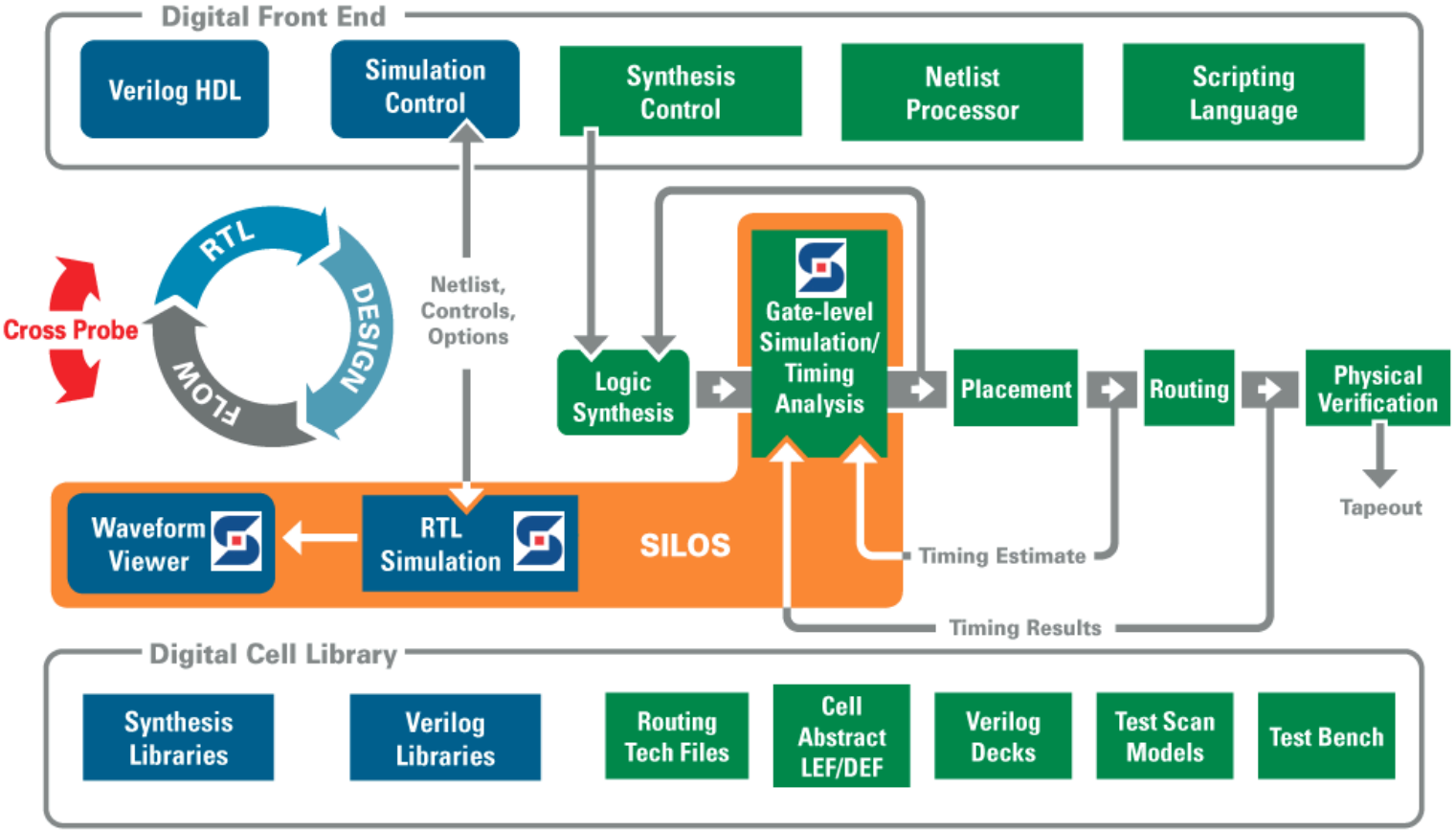
SILVACO

What is Silos

- Verilog(IEEE 1364) Digital Logic Simulator
  - Analysis and Debugging Environment
    - Waveform Viewer
    - Hierarchical Browser
    - Text editor
    - Code Coverage
    - Finite State Machine Tool
    - Single Step with Breakpoints
    - Trace Gate and RTL events

SILVACO

Digital Flow



SILVACO

Silos Platforms

- Silos supports
  - Windows
  - SUN Solaris
  - Linux platforms

SILVACO

Silos Markets

- Current
  - FPGA (Xilinx, Altera, QuickLogic, Actel, etc.)
  - PLD
  - General Digital Logic
- Future
  - Analog Mixed-Signal

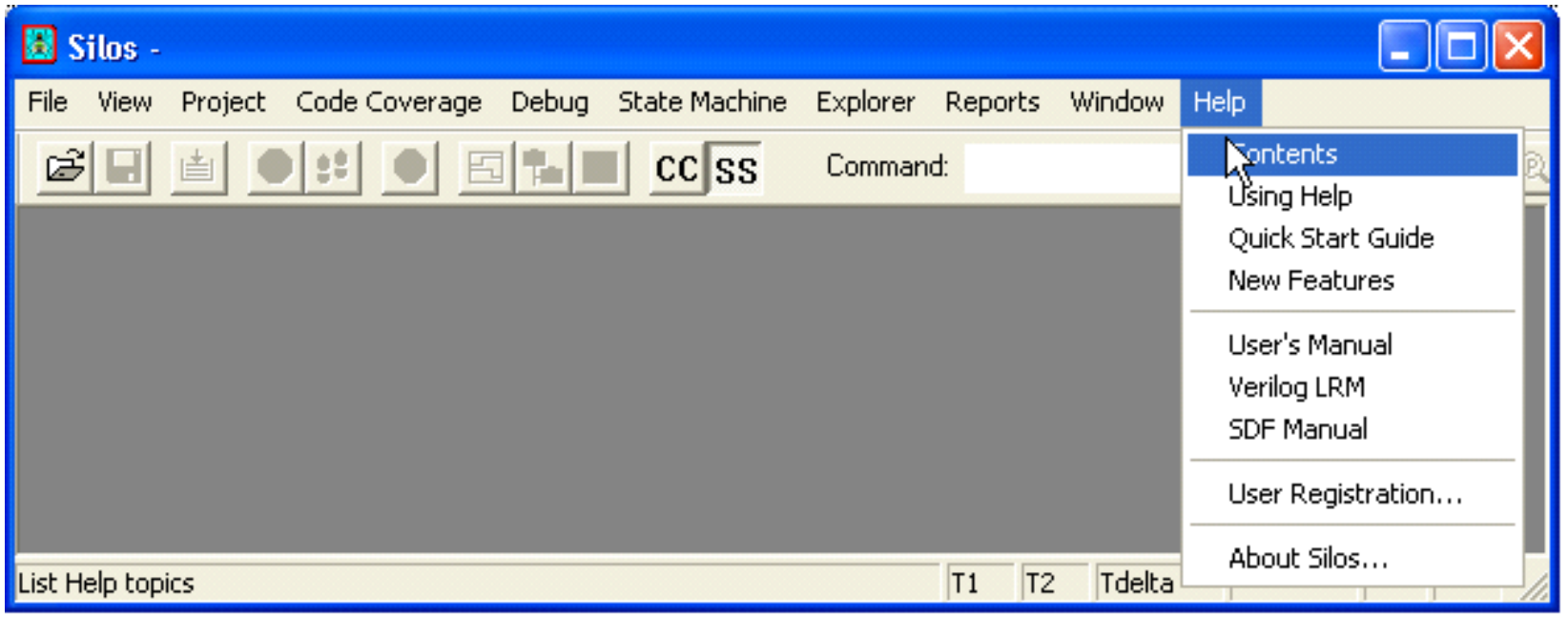
SILVACO

Agenda: Starting Silos Project

- Introduction
- Starting Silos Project
- Explorer and Analyzer
- Source Code Debugging
- State Machine Design Entry
- Advanced Debugging Features
- Finite State Machine Example
- Gate Level Debugging

SILVACO

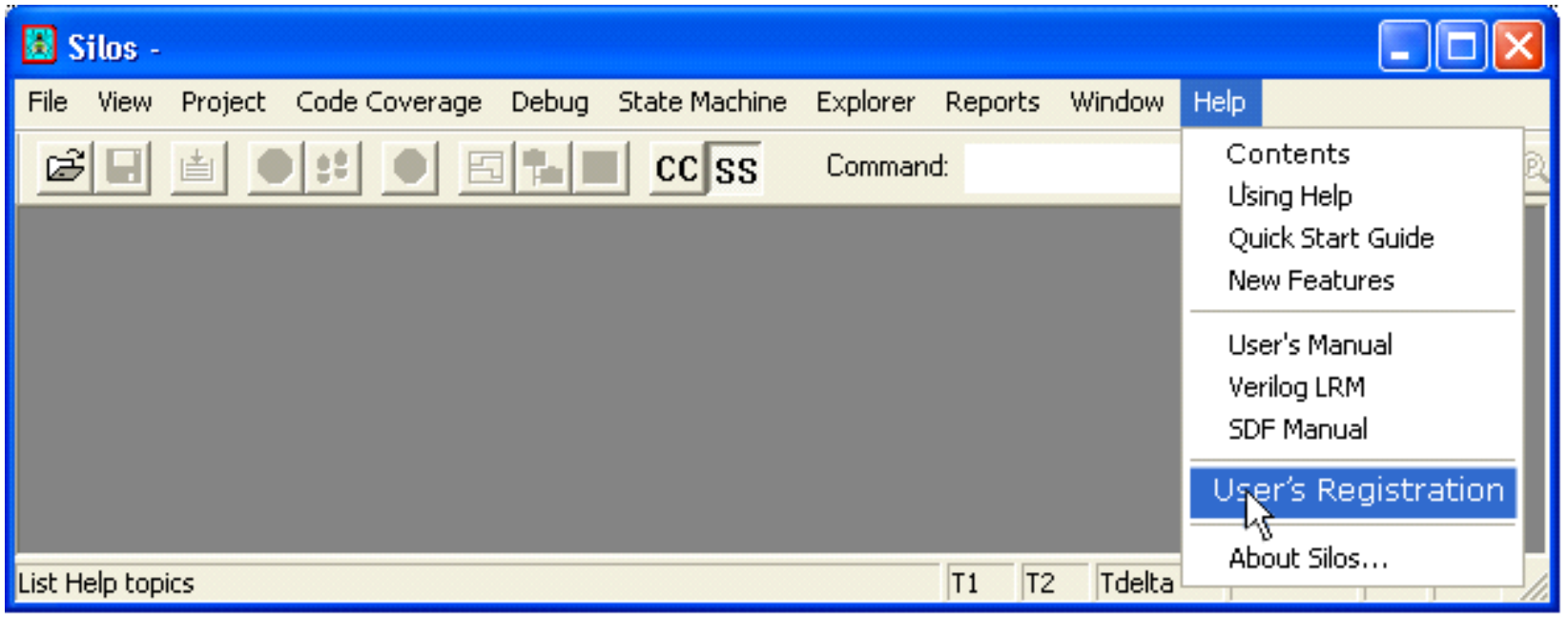
Starting Silos



"Contents" will display the contents for the Silos User's Manual. The "New Features" menu provides a short help file on the new features for the Silos release.

SILVACO

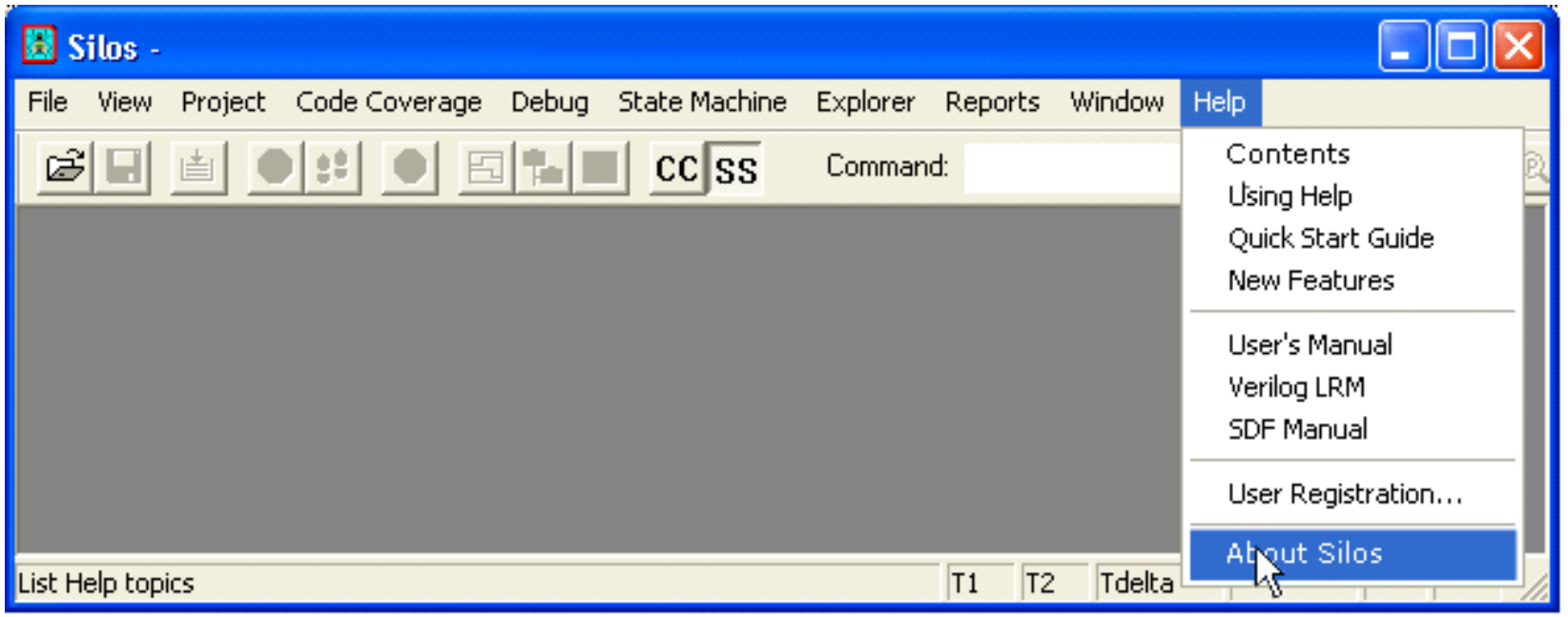
Starting Silos



"User Registration" will show the security block number.

SILVACO

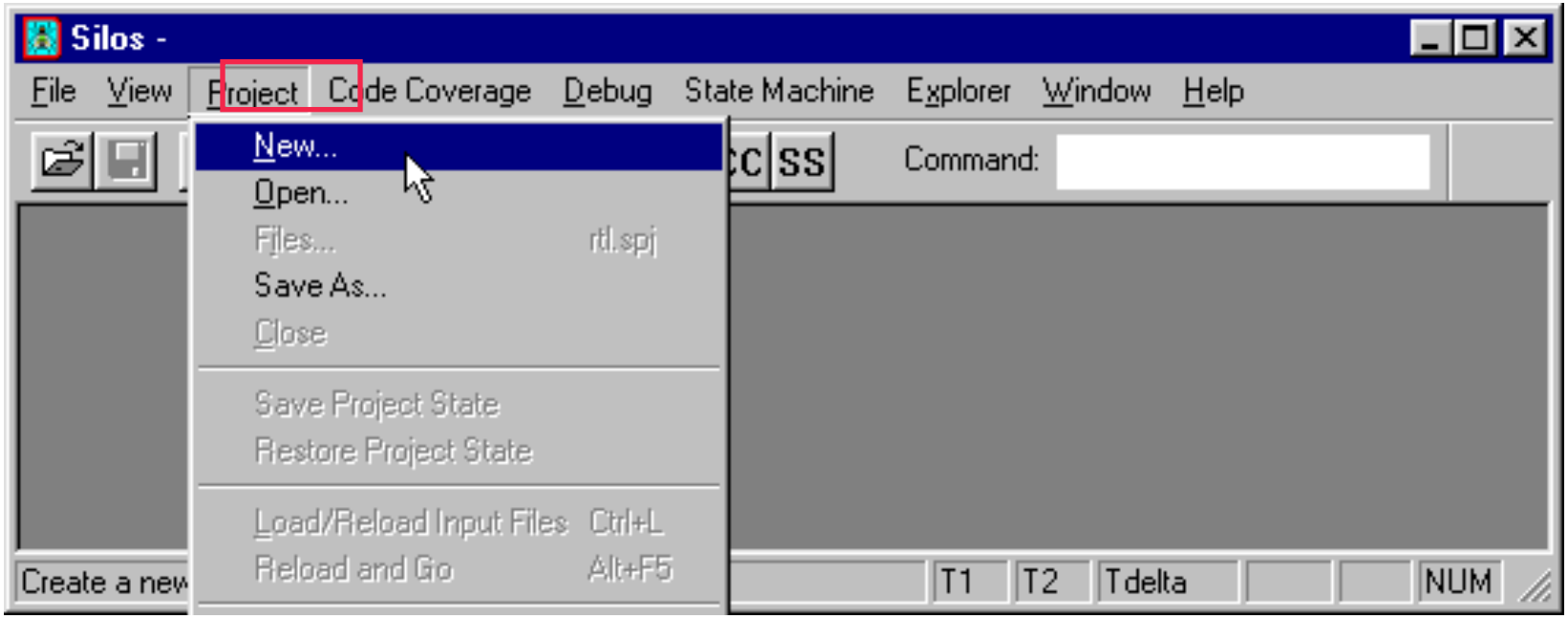
Starting Silos



"About Silos" will show the memory usage and version number.

SILVACO

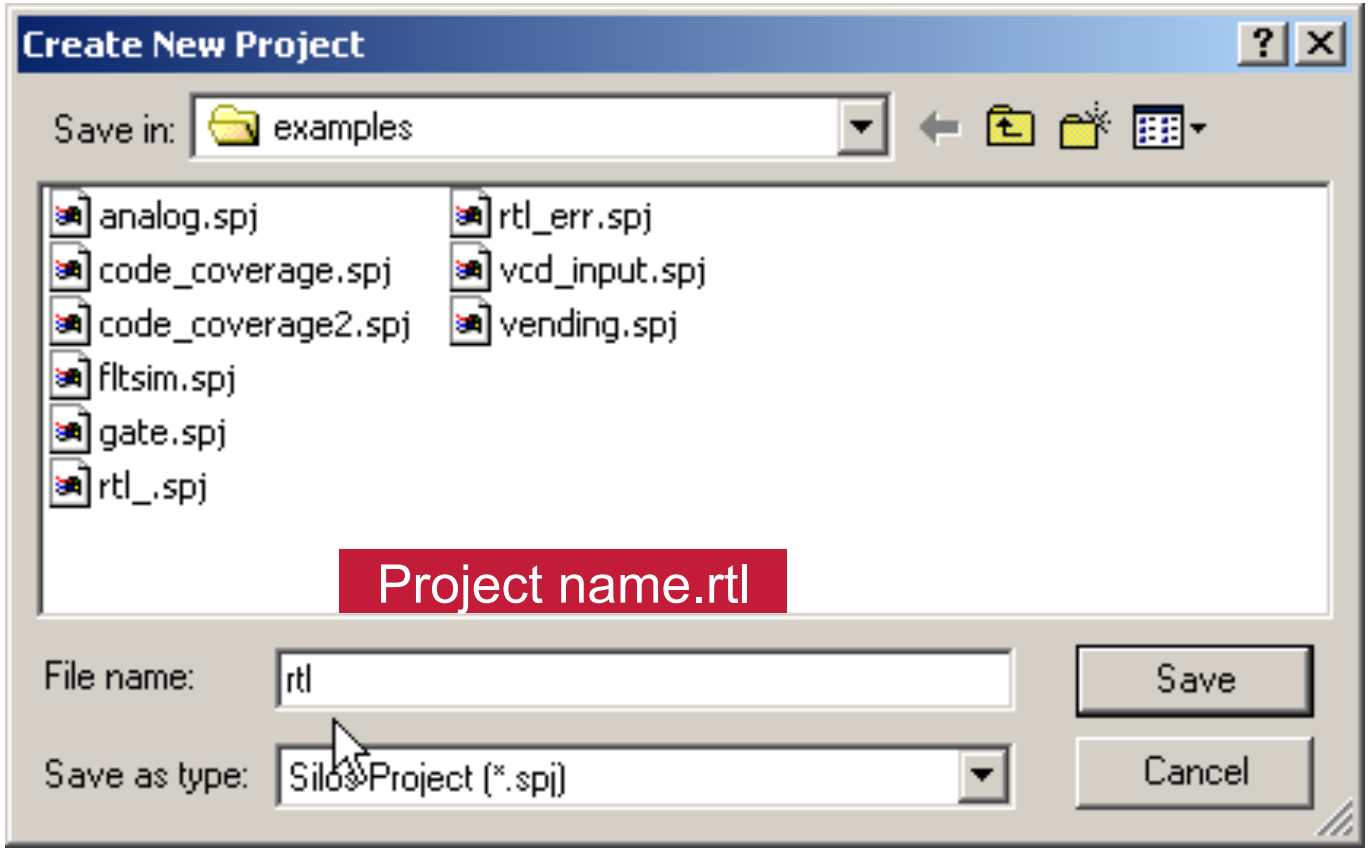
Opening a Project



Highlight “Project/New” and press and release the “F1” key on the keyboard to see the on- line help.

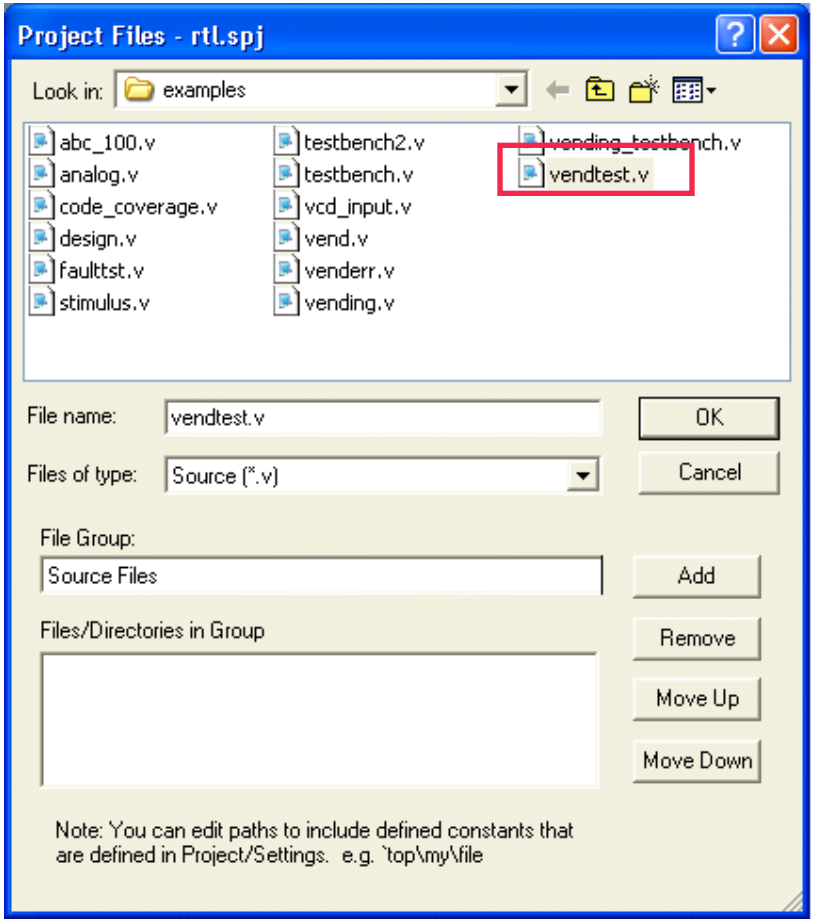
SILVACO

Create a Project



SILVACO

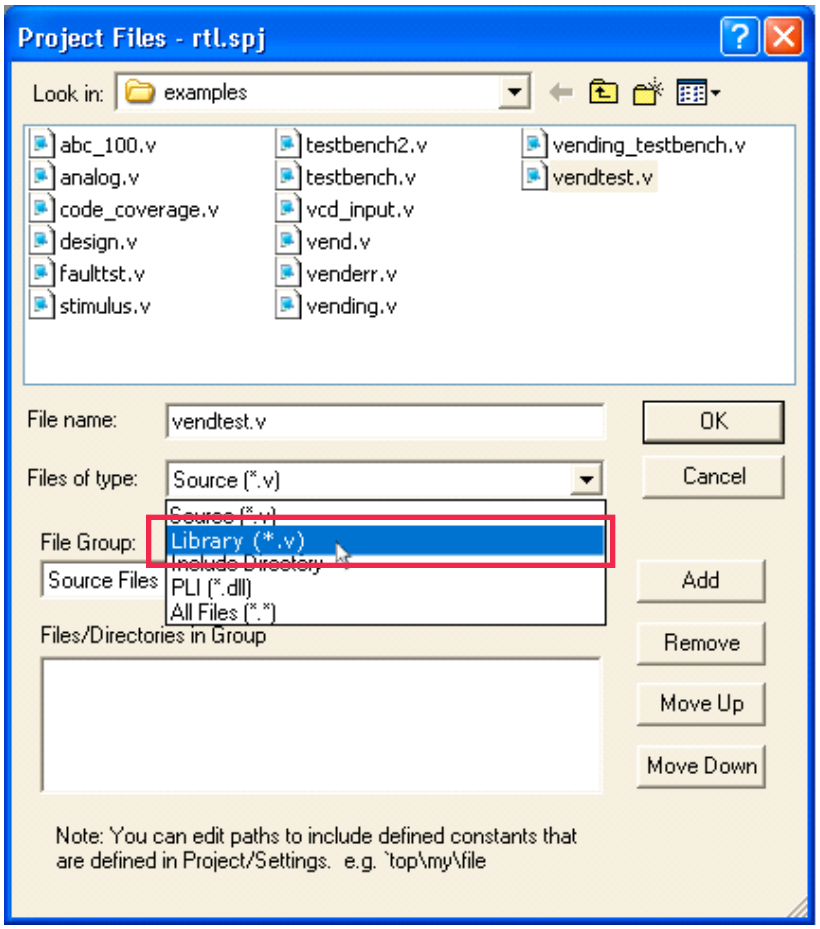
Project Files



Double click on a file name to add it to project.

SILVACO

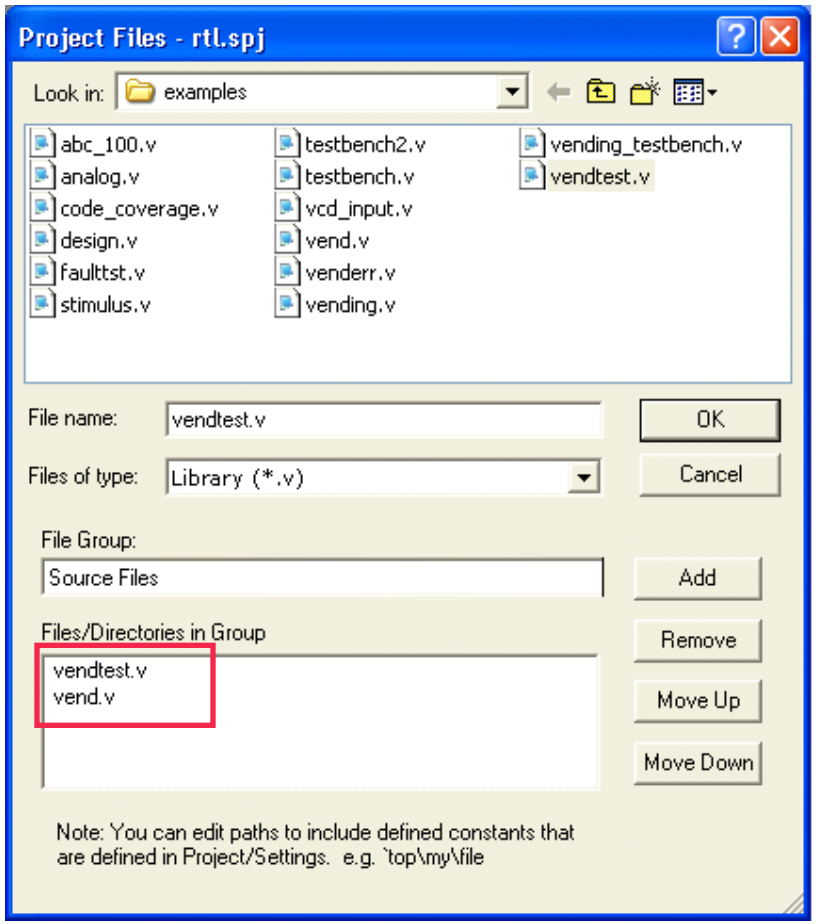
Project Files



Drop down arrow for library files.

SILVACO

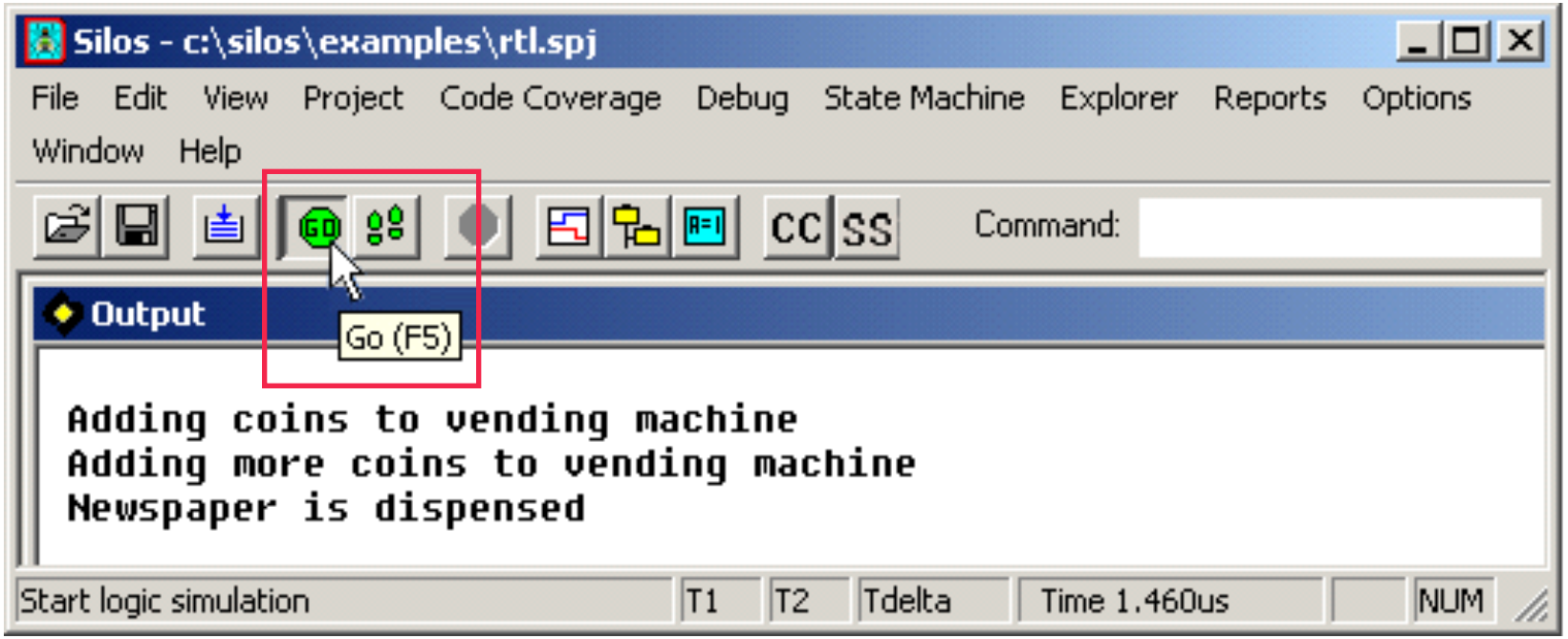
Project Files



Input files for the design.

SILVACO

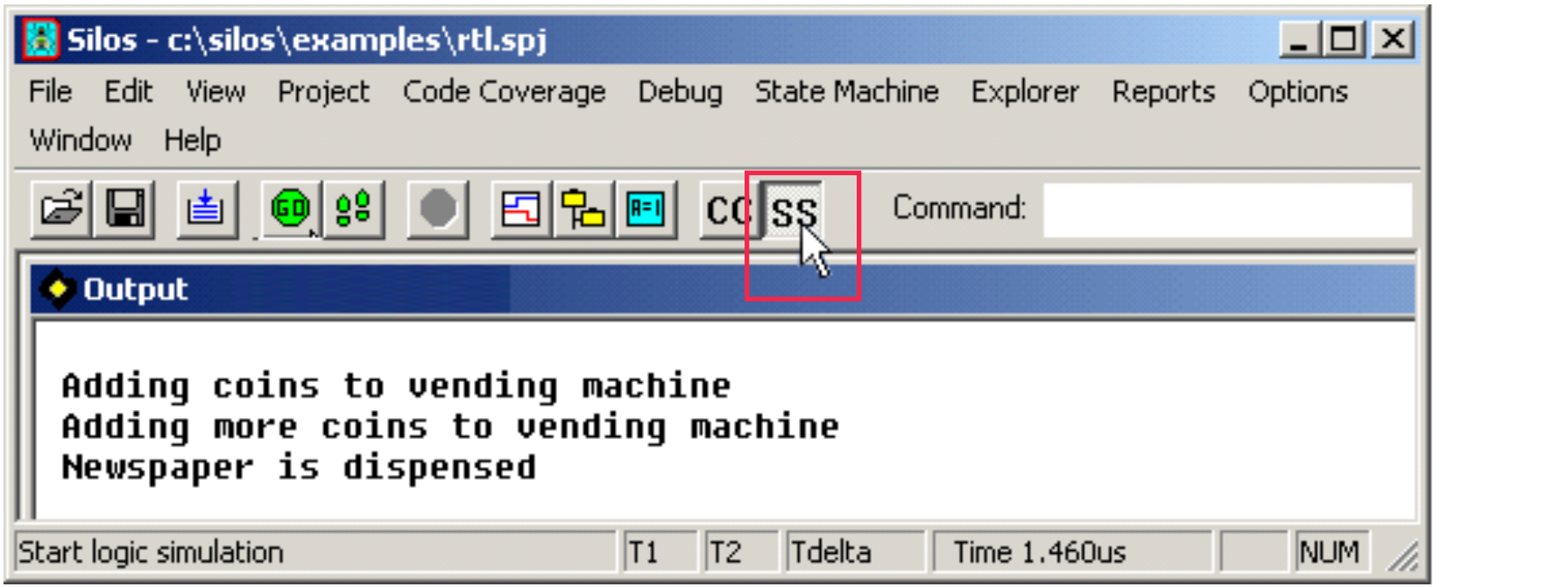
Starting a Simulation



Press “Go” button (F5)

SILVACO

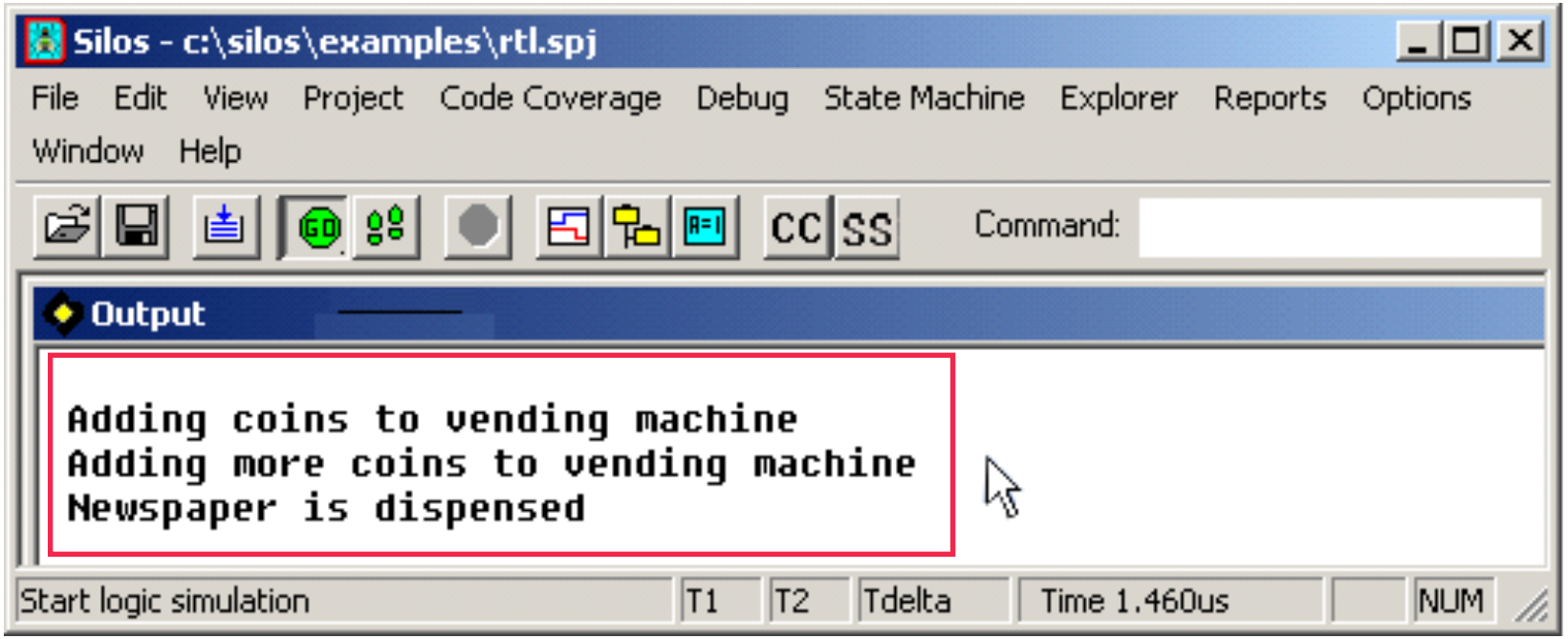
Starting a Simulation



SS Button for Singe Stepping.

SILVACO

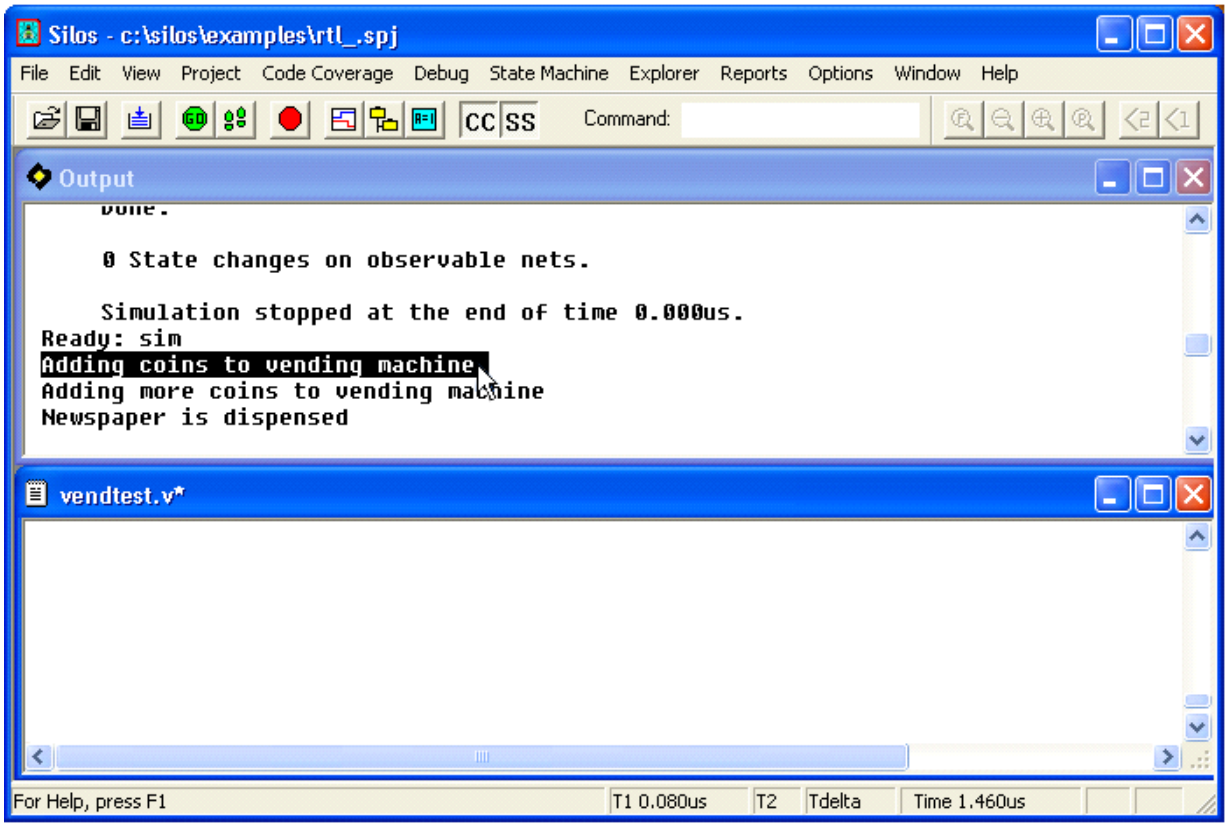
Starting a Simulation



Output Window.

SILVACO

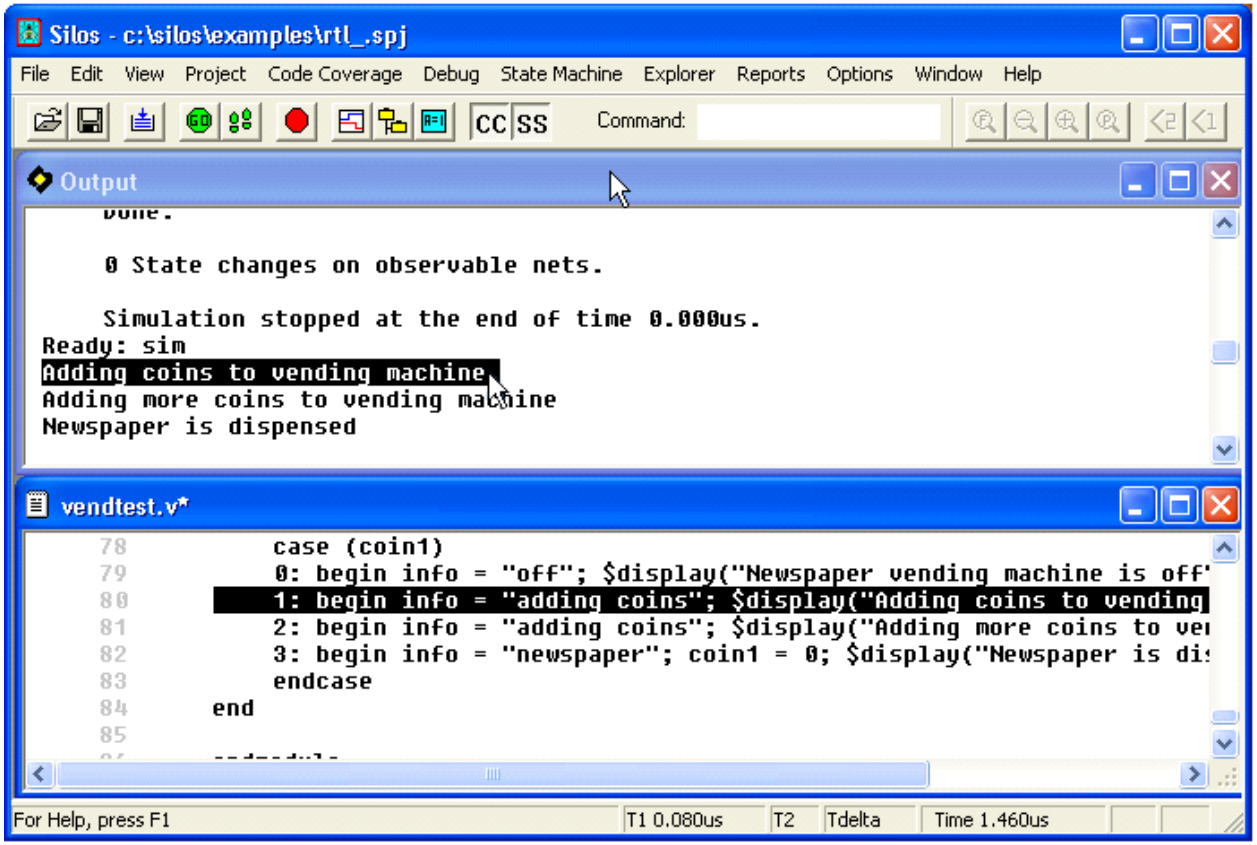
Output of a Simulation



Double clicking on the text in the Output window will open the source file…

SILVACO

Output of a Simulation


Silos - c:\silos\examples\rtl_.spj
File Edit View Project Code Coverage Debug State Machine Explorer Reports Options Window Help
Command:
Output
0 State changes on observable nets.
Simulation stopped at the end of time 0.000us.
Ready: sim
Adding coins to vending machine
Adding more coins to vending machine
Newspaper is dispensed
vendtest.v*
78 case (coin1)
79 0: begin info = "off"; $display("Newspaper vending machine is off"
80 1: begin info = "adding coins"; $display("Adding coins to vending
81 2: begin info = "adding coins"; $display("Adding more coins to ve
82 3: begin info = "newspaper"; coin1 = 0; $display("Newspaper is di
83 endcase
84 end
85
For Help, press F1
T1 0.080us
T2
Tdelta
Time 1.460us

…and highlight the $display statement that caused the text to be printed.

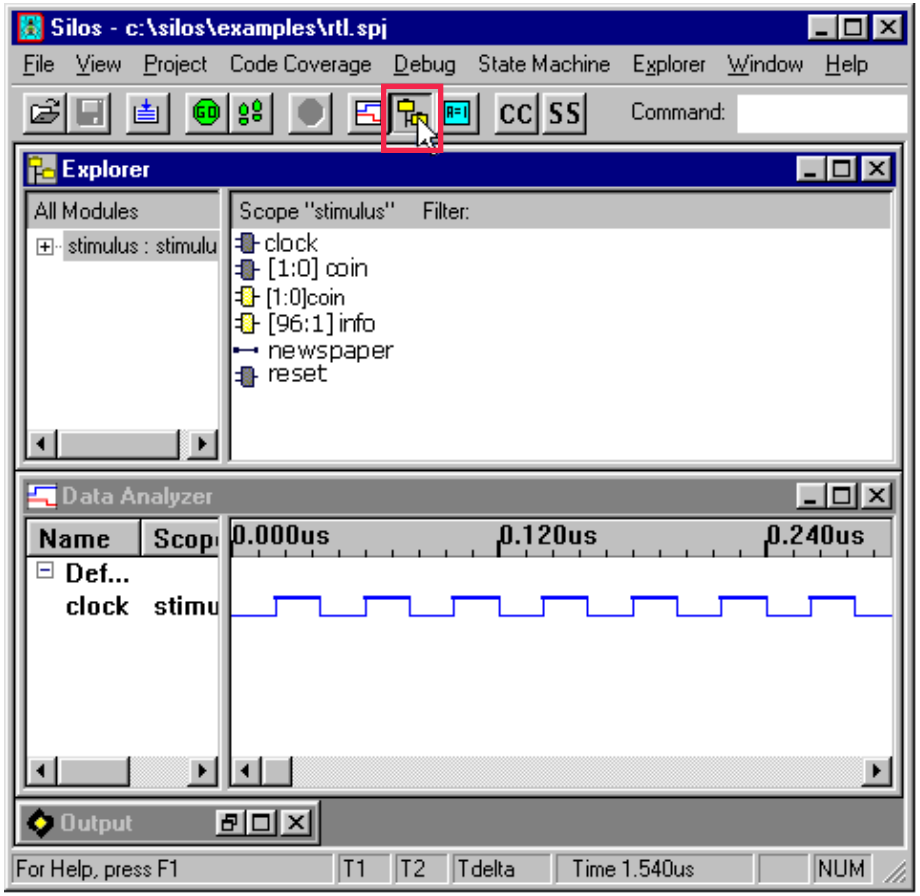
SILVACO

Agenda: Explorer and Analyzer

- Introduction
- Starting Silos Project
- Explorer and Analyzer
- Source Code Debugging
- State Machine Design Entry
- Advanced Debugging Features
- Finite State Machine Example
- Gate Level Debugging

SILVACO

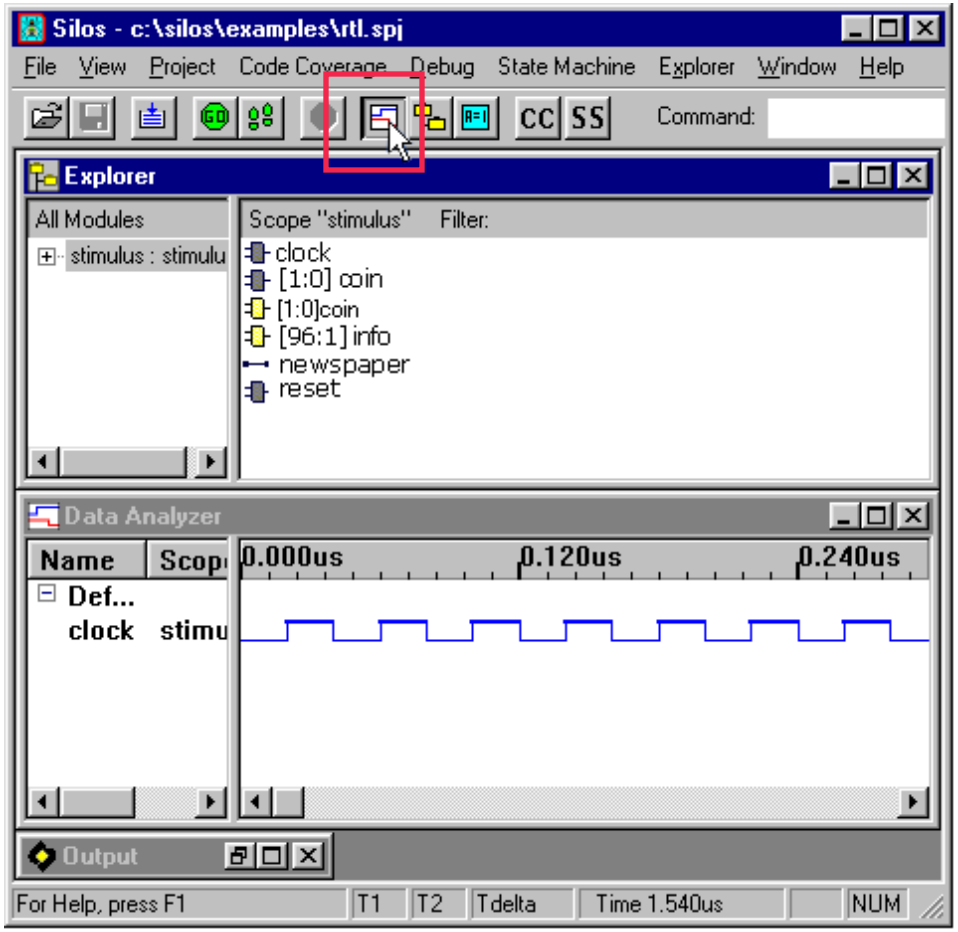
Explorer and Analyzer



Open Explorer button.

SILVACO

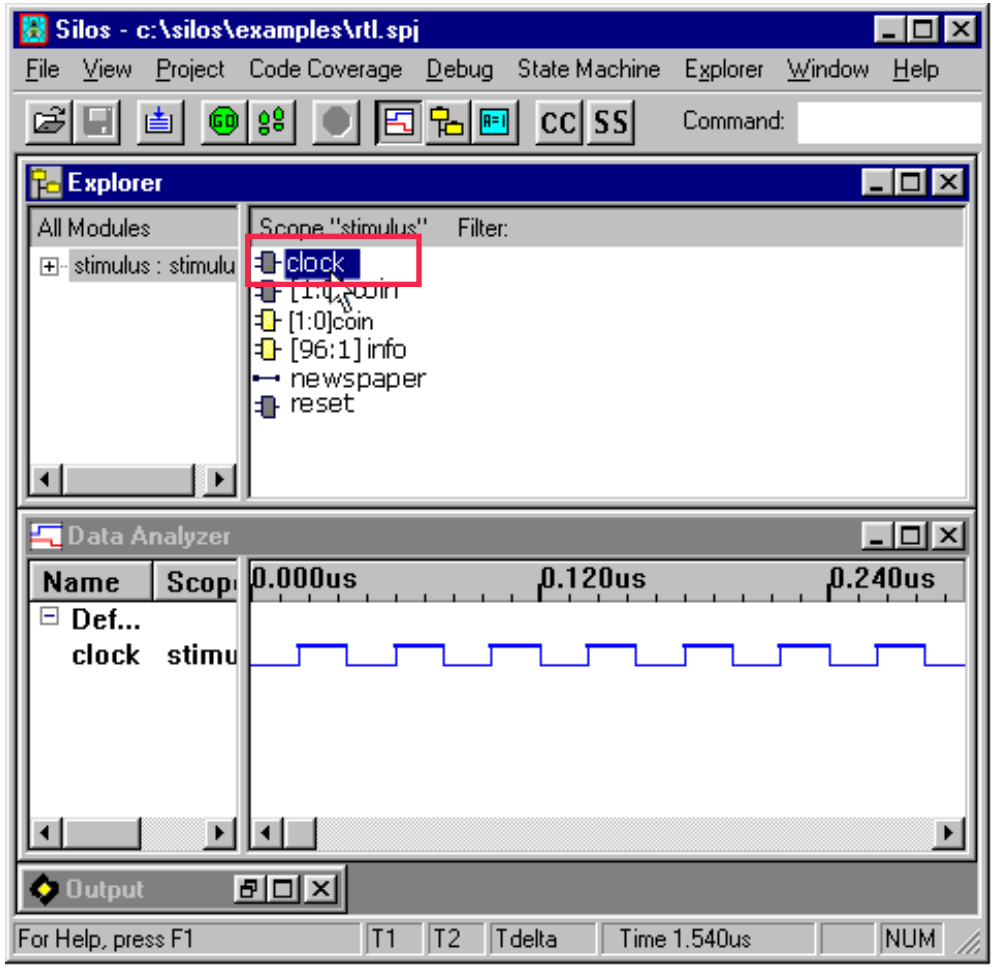
Explorer and Analyzer



Open Analyzer button.

SILVACO

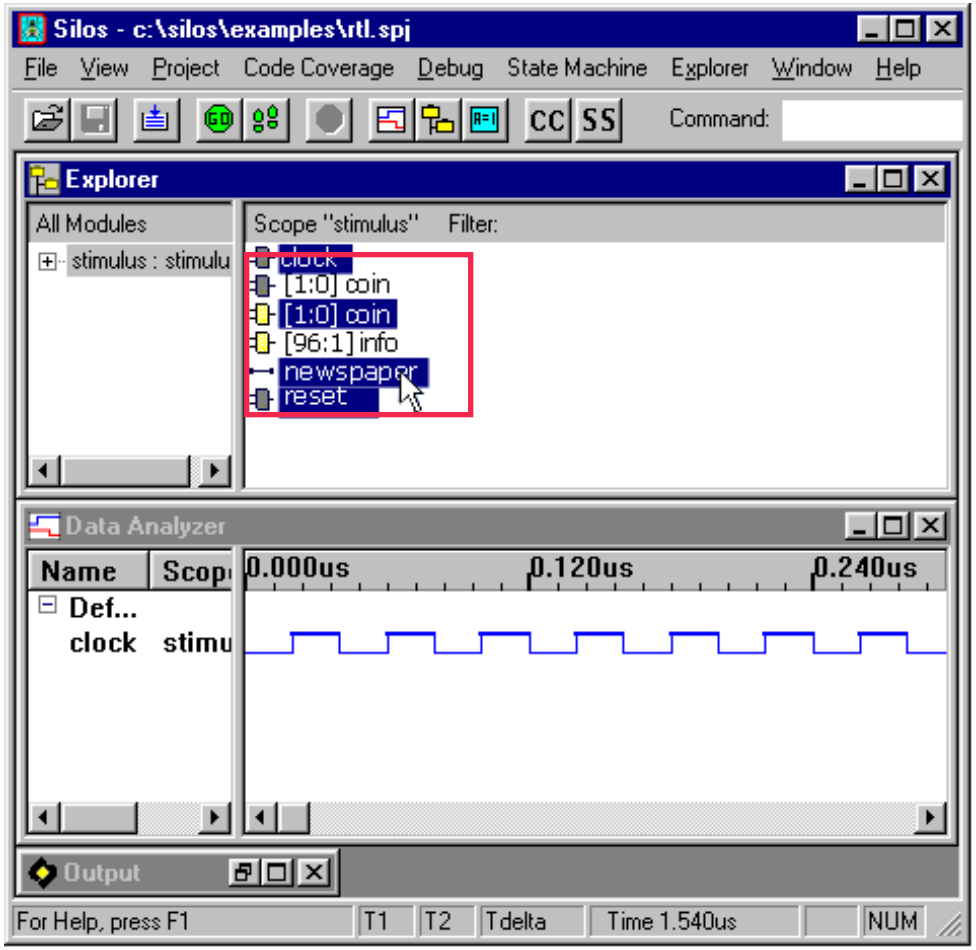
Explorer and Analyzer



Select signal “clock”.

SILVACO

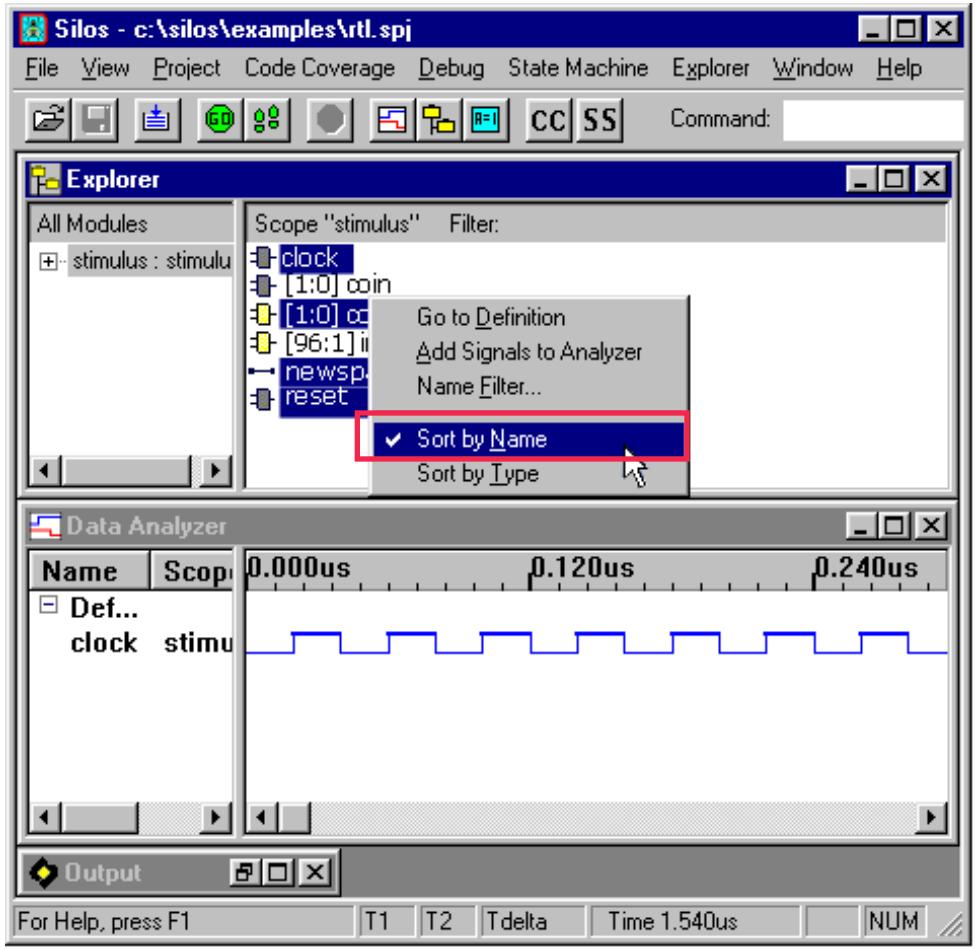
Explorer and Analyzer



Hold down Ctrl or Shift keys while using the mouse to select additional signals.

SILVACO

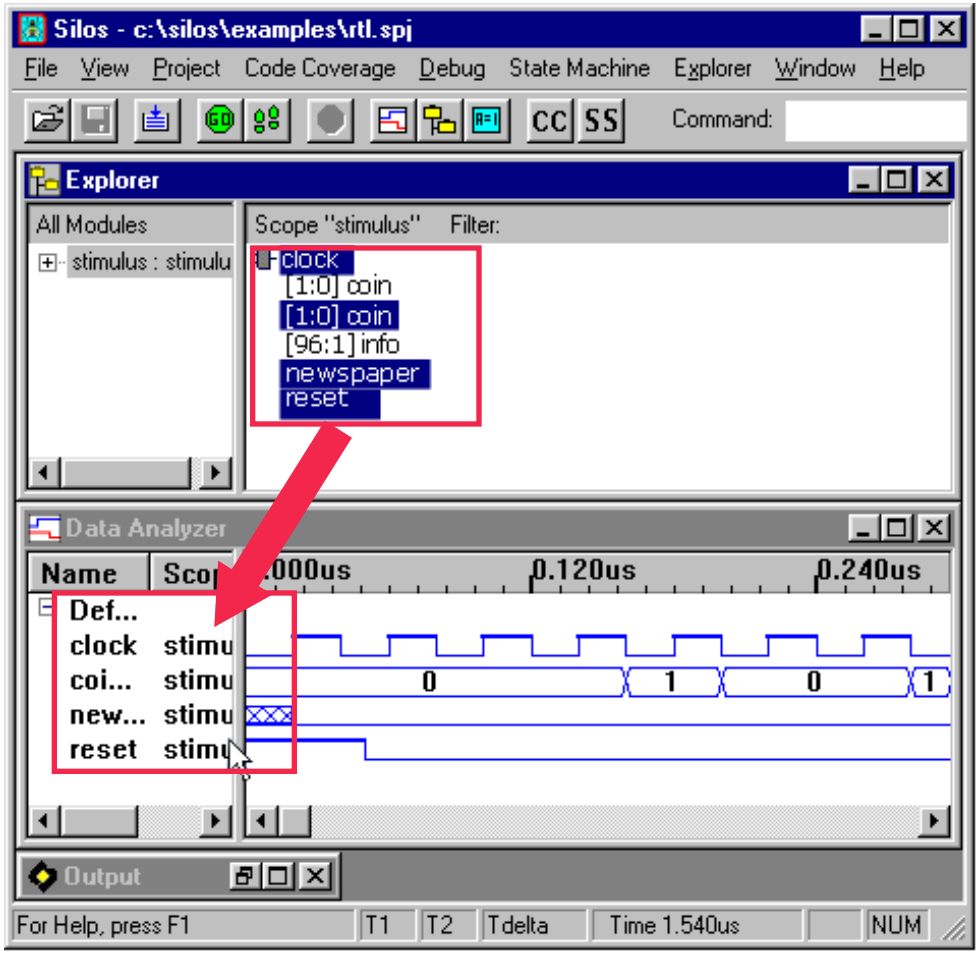
Explorer and Analyzer



Click with the right mouse button to open the context menu and select “Sort by Name”.

SILVACO

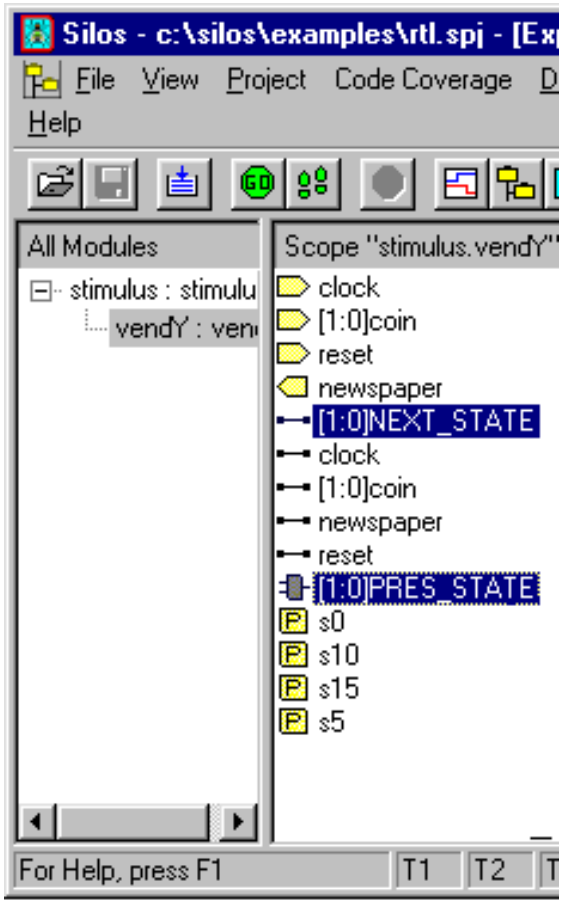
Explorer and Analyzer



Drag and drop signals to "Name" list box in Data Analyzer.

SILVACO

Explorer - Symbols



Input ports point right.

Output ports point left.

Non-port variable symbols:

"dumbell" symbol is a wire

"flop" symbol is a register

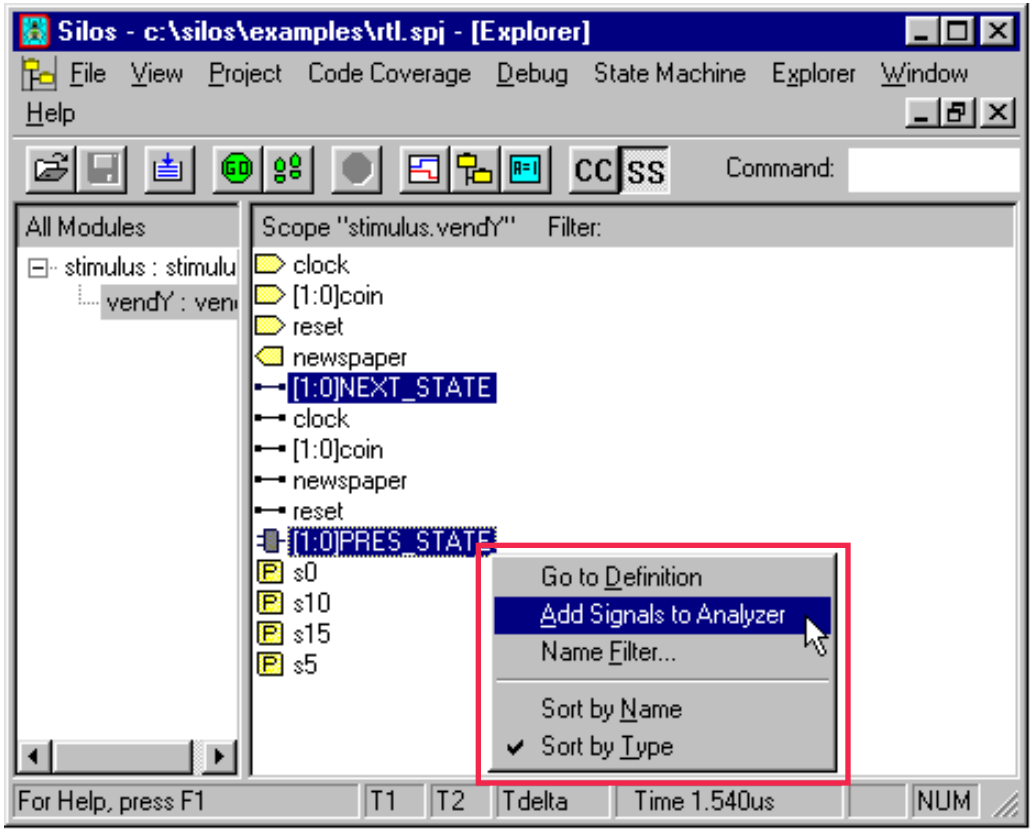
is a real variable

is a parameter

is an integer

SILVACO

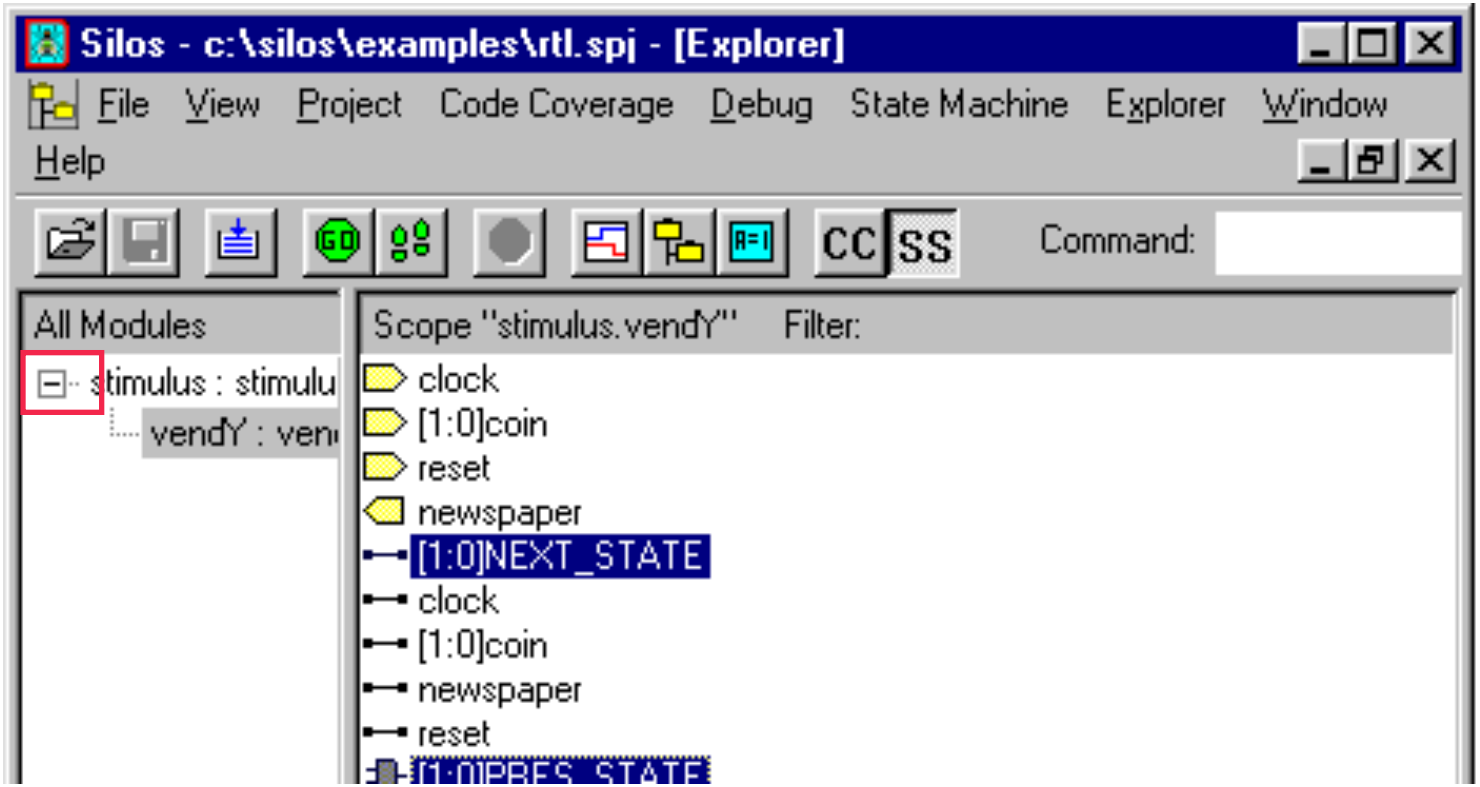
Explorer - Context Menu



Context menu (use right mouse button).

SILVACO

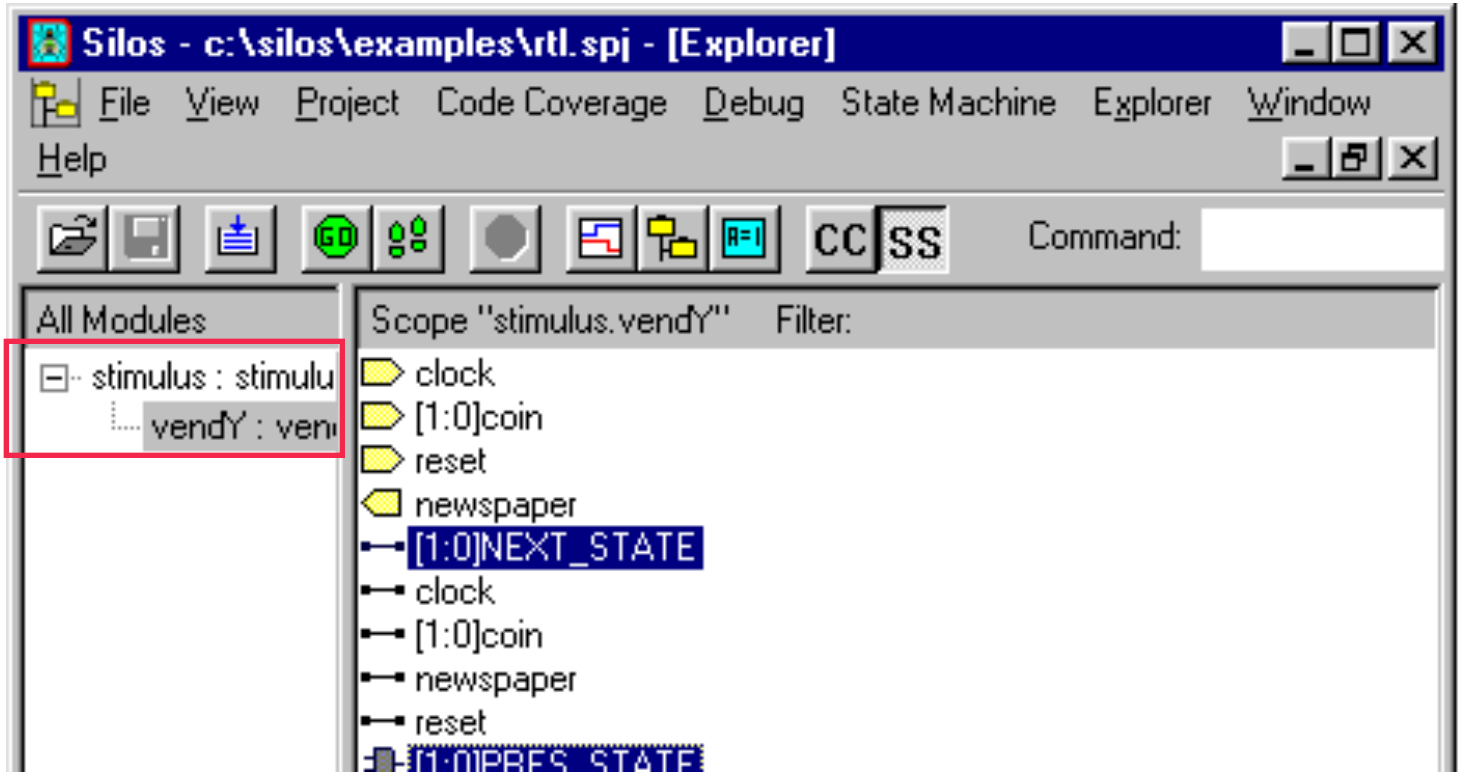
Explorer



Click here to expand and contract hierarchy.

SILVACO

Explorer



Use right mouse button to see the context menu.

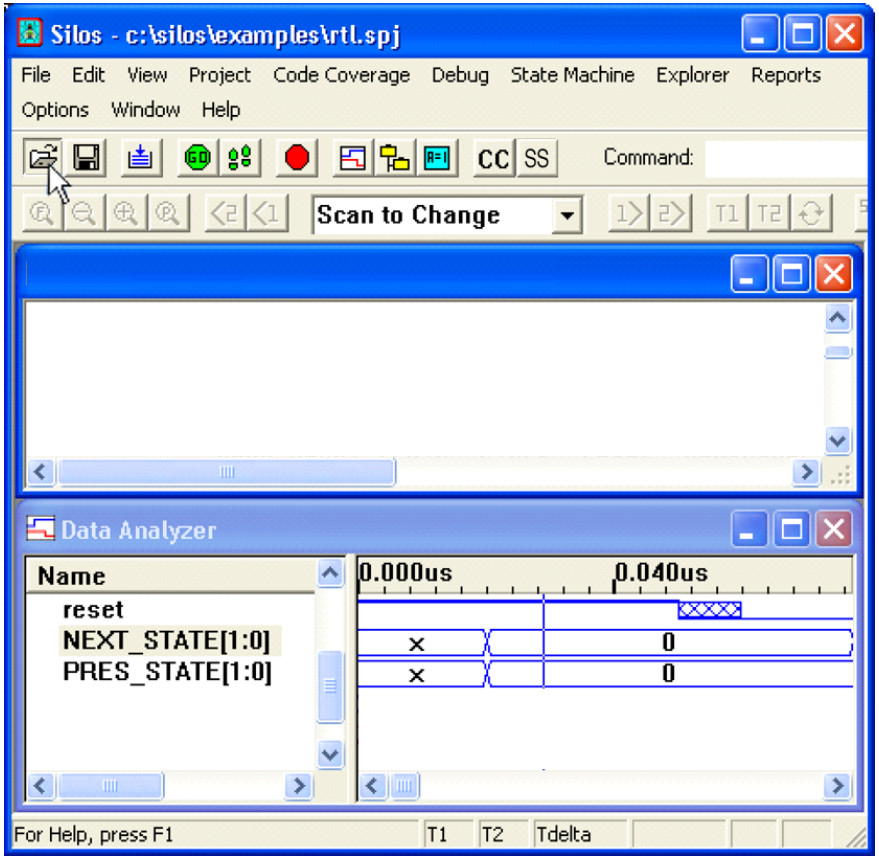
SILVACO

Agenda: Source Code Debugging

- Introduction
- Starting Silos Project
- Explorer and Analyzer
- Source Code Debugging
- State Machine Design Entry
- Advanced Debugging Features
- Finite State Machine Example
- Gate Level Debugging

SILVACO

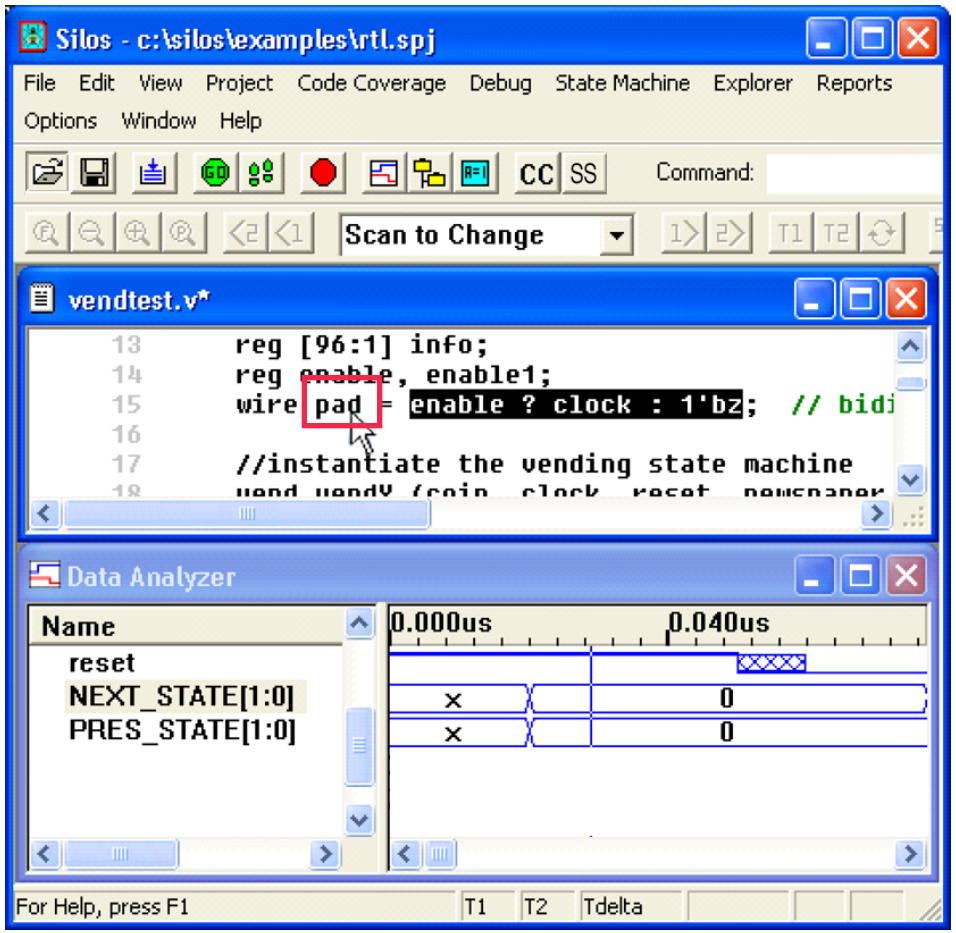
Source Code Waveforms



"Open File" button to open file "vendtest.v".

SILVACO

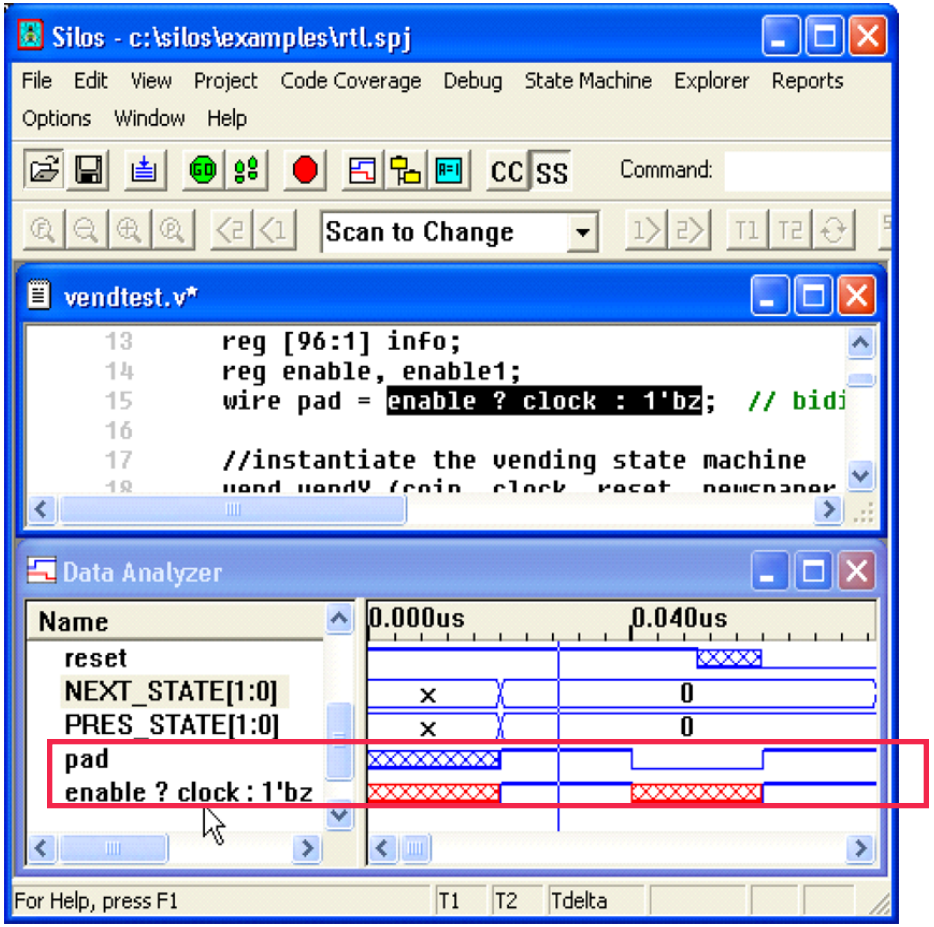
Source Code Waveforms



Double click on variable "pad" and drag and drop "pad" into the Name column to see the waveform.

SILVACO

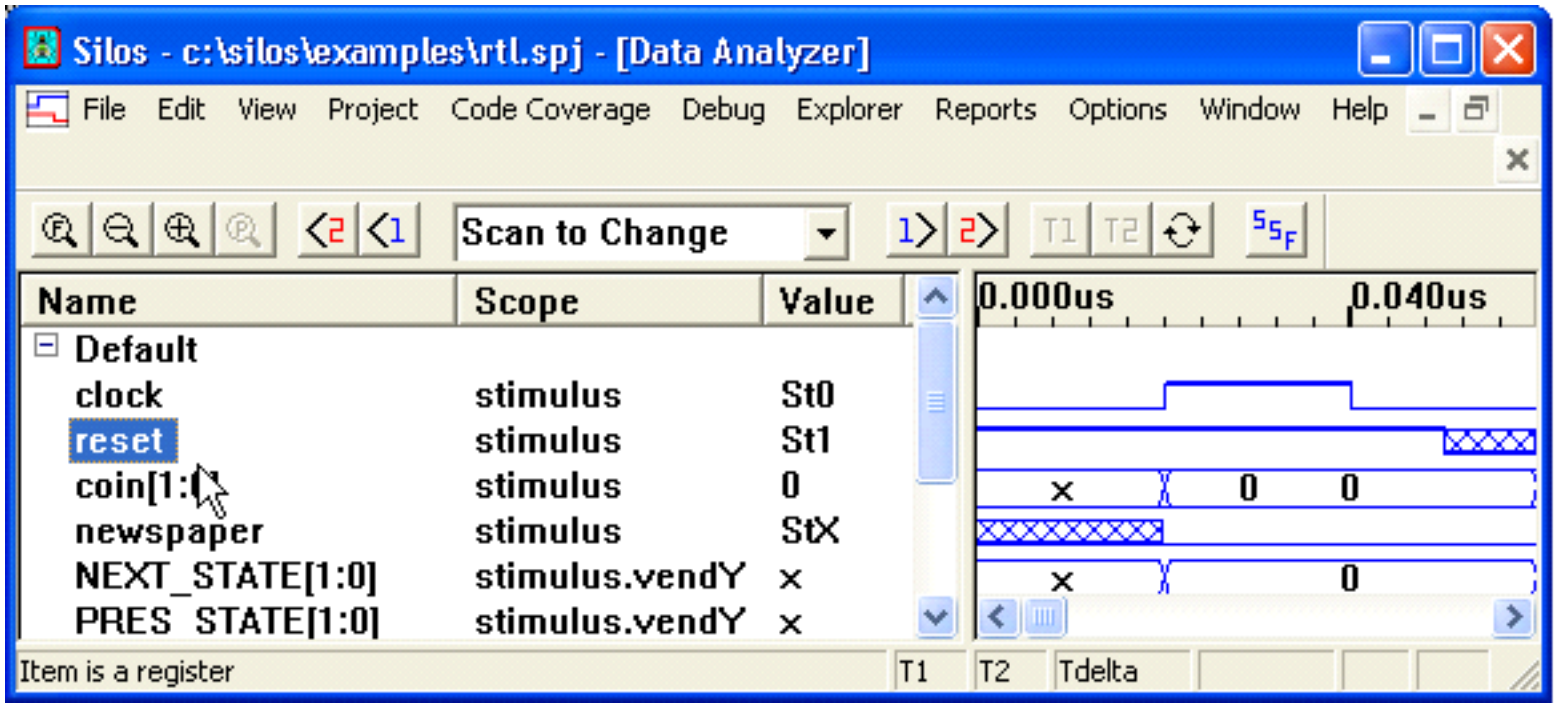
Source Code Waveforms



Highlight expression "enable ? clock : 1'bz" and drag and drop it into the Data Analyzer to see the waveform for the expression.

SILVACO

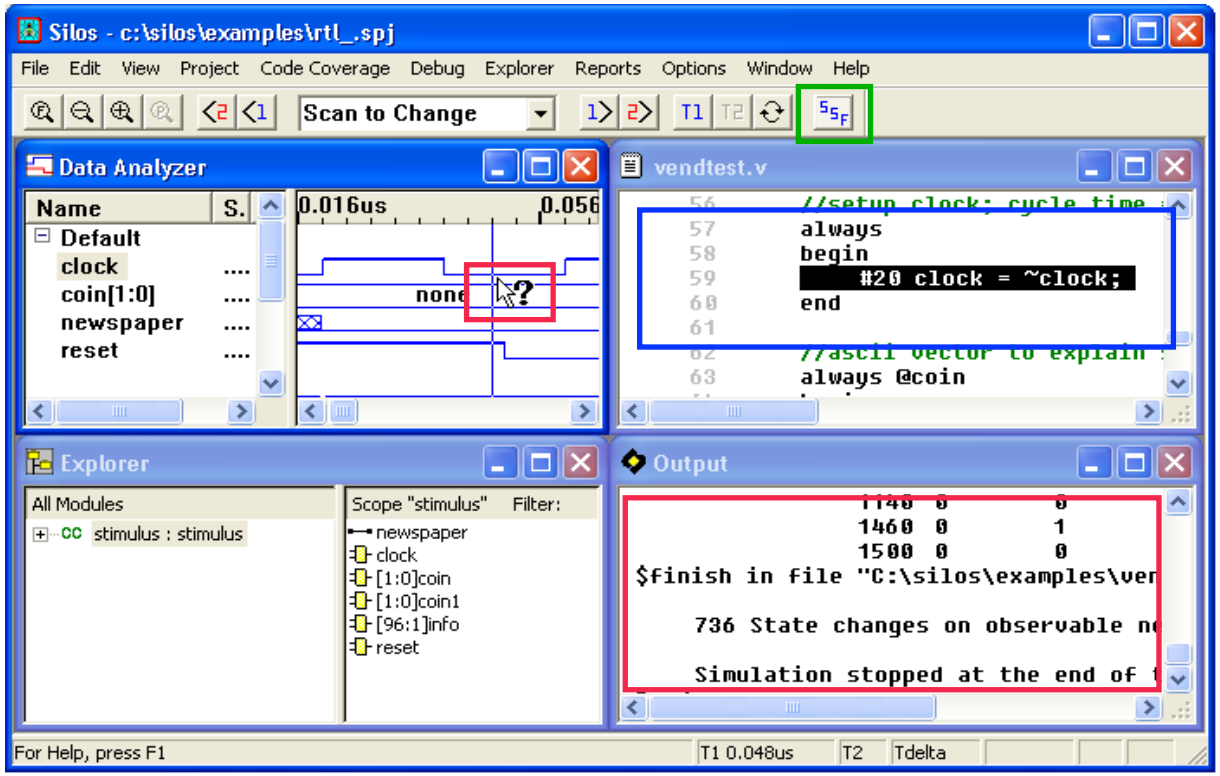
Rearranging Signal – Drag and Drop



Rearrange the signal names by dragging and dropping them in the Signal Name list box.

SILVACO

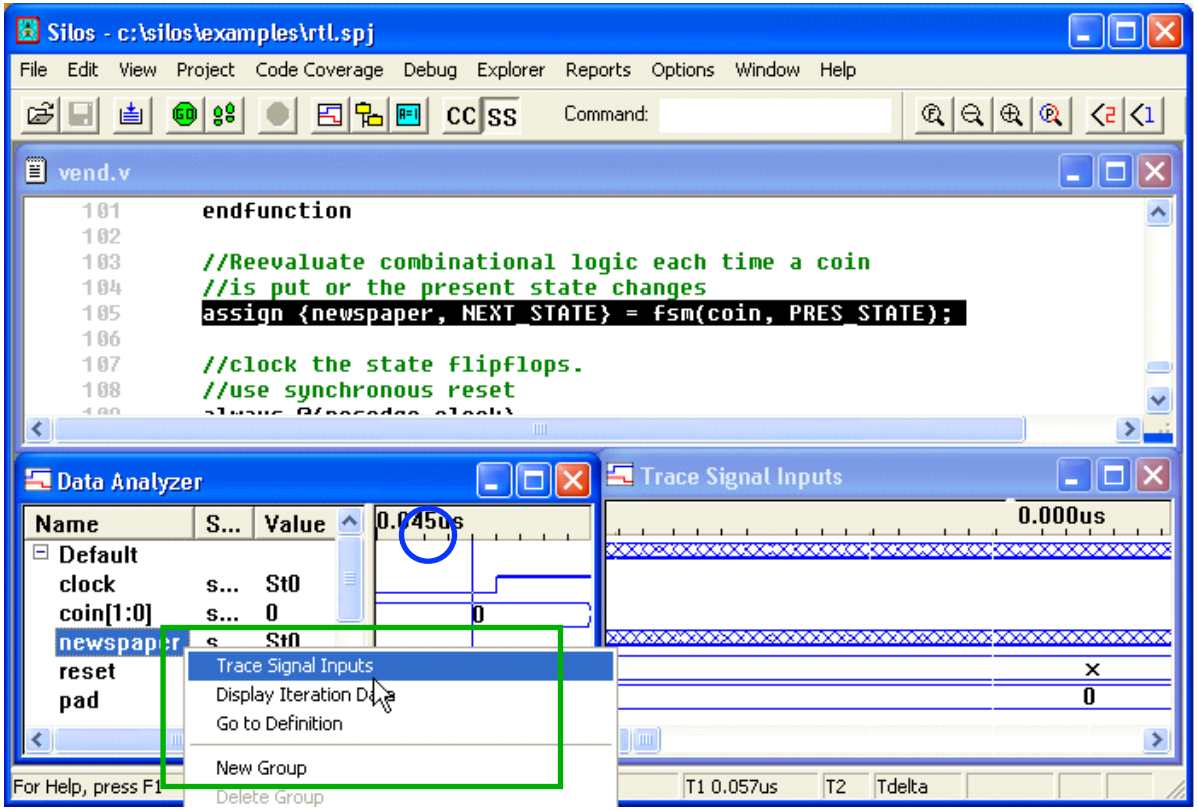
Show Source Code



If “Show Source” in inactive (SSF button not depressed) [1], holding down the “Ctrl” key will temporarily activate Show Source [2]. You can then left mouse click on the waveform for “clock” to see the statement that caused “clock” to toggle [3].

SILVACO

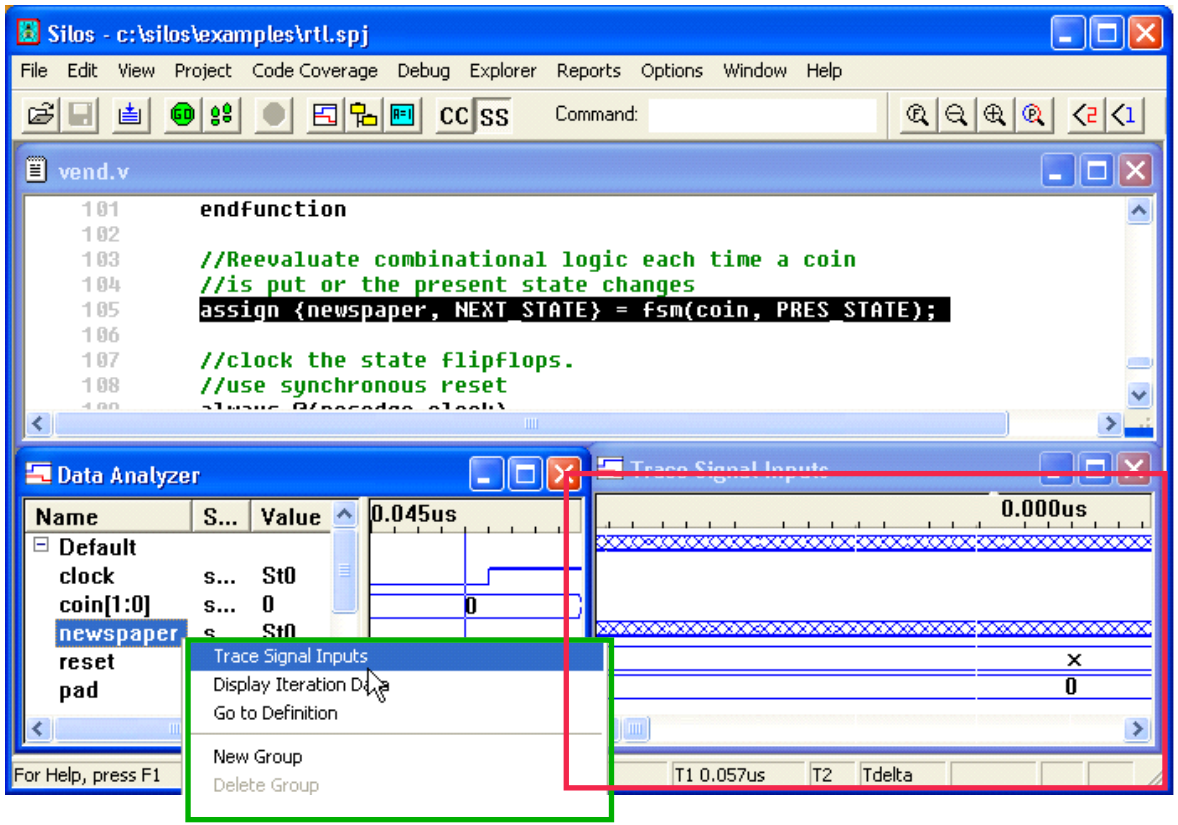
Visual Debug - Time Point for Tracing Signal



Setting the blue “T1” marker [1] determines the time point for tracing a signal.
Right mouse clicking on the name “newspaper” brings up the context menu [2].

SILVACO

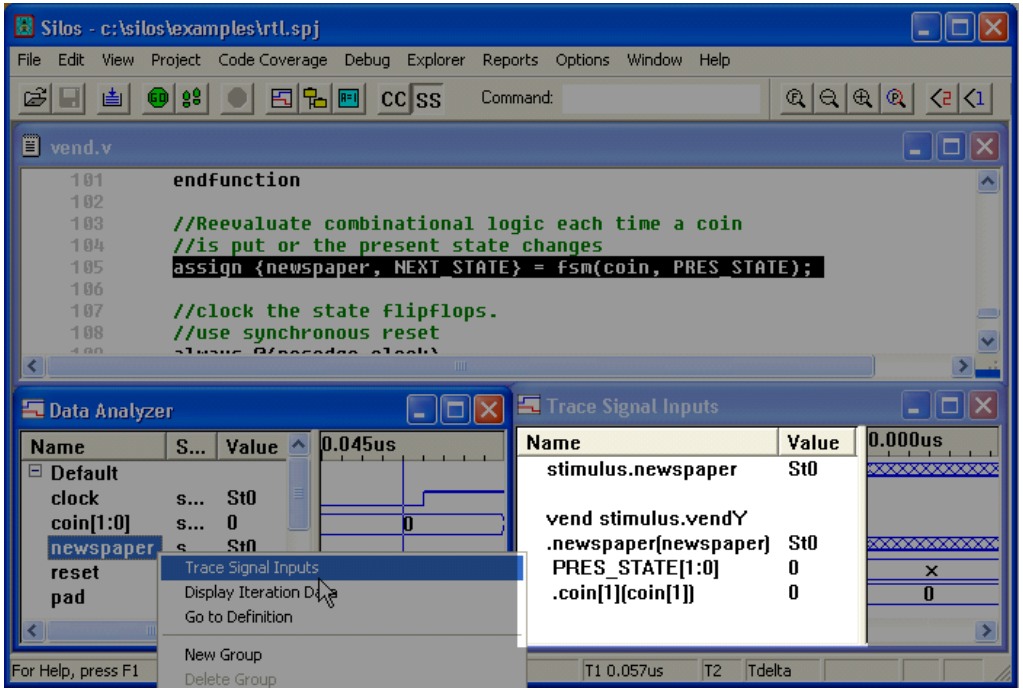
Visual Debug - Trace Signal Inputs



Selecting the Trace Signal Inputs menu selection opens the Trace Signal Inputs window [3].

SILVACO

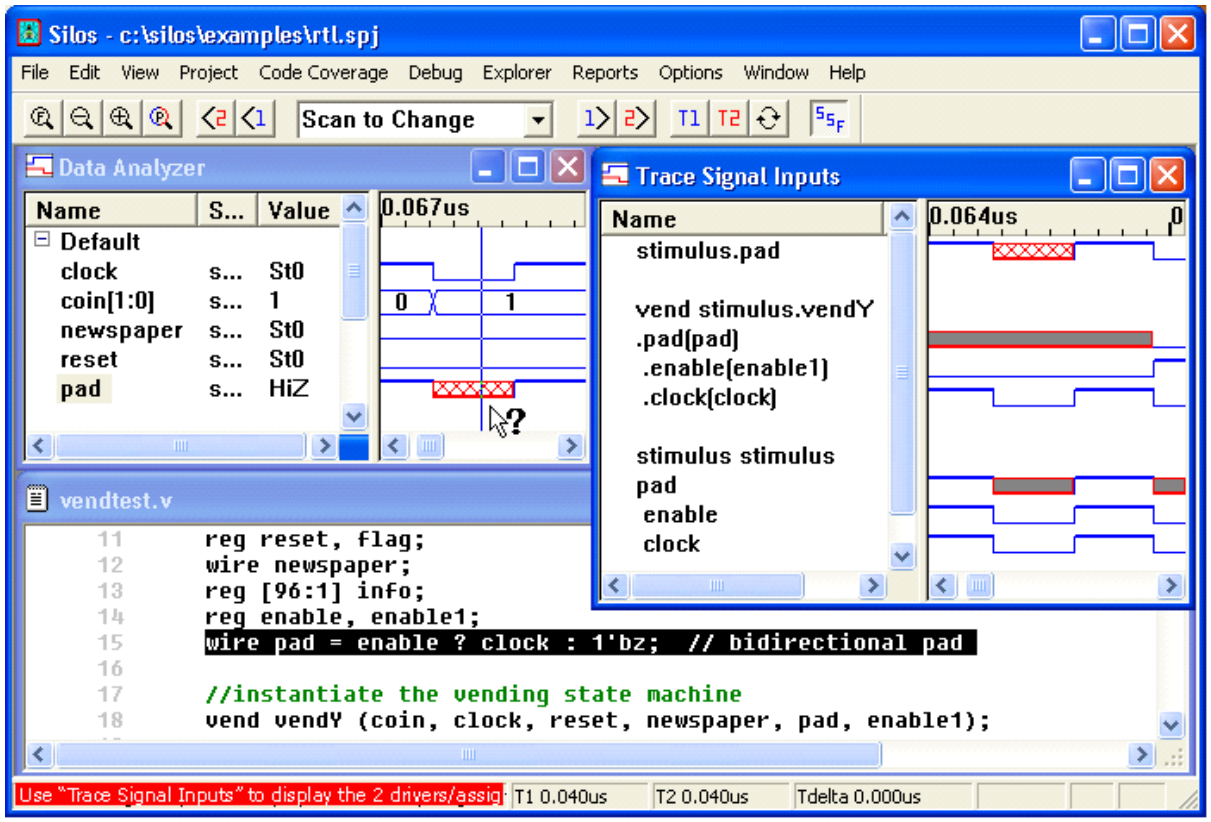
Visual Debug



The top signal, “stimulus newspaper”, is the signal being traced. The next line is the module name “vend”, and instance name, “stimulus.vendY”, which drives the top signal. The next line is the driver, or assignment, in the instance. Subsequent lines indented with a space are the inputs or right hand side (rhs) variables to the driver/assignment. If the driver or rhs variable is a port of the instance it is shown using port name syntax. If it is not a port then just the name of the variable is listed. The signal name preceded by the period, “.newspaper”, is the local port name in module “vend”. The name in the parenthesis, “(newspaper)”, is the name in the module above “stimulus.vendY”. Signal names “coin[1]” and wire “PRES_STATE[1:0]”, the rhs variables in the assignment to “newspaper”, are listed below it and indented one space.

SILVACO

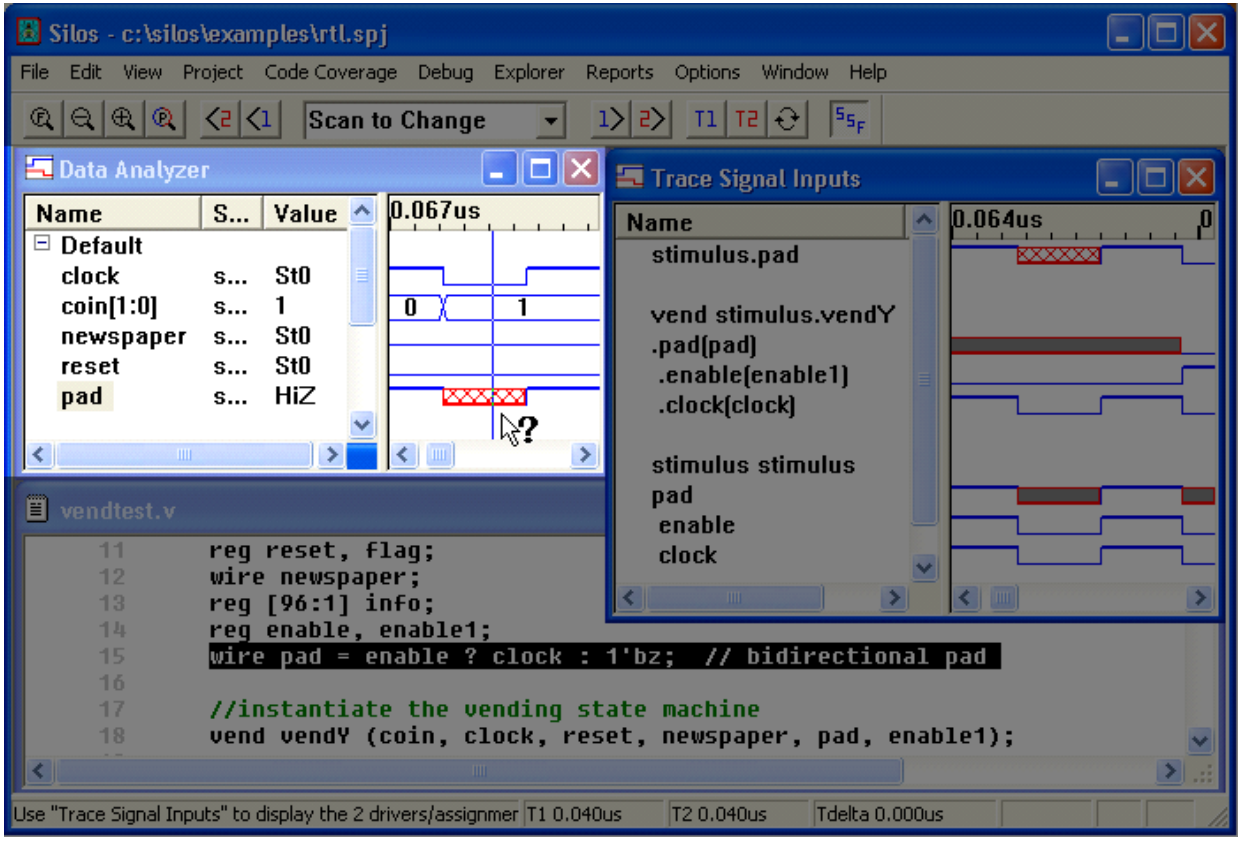
Visual Debug - Status Window



Red flashing message in status window warns user of multiple drivers or assignments at a time point while “Show Source” is on.

SILVACO

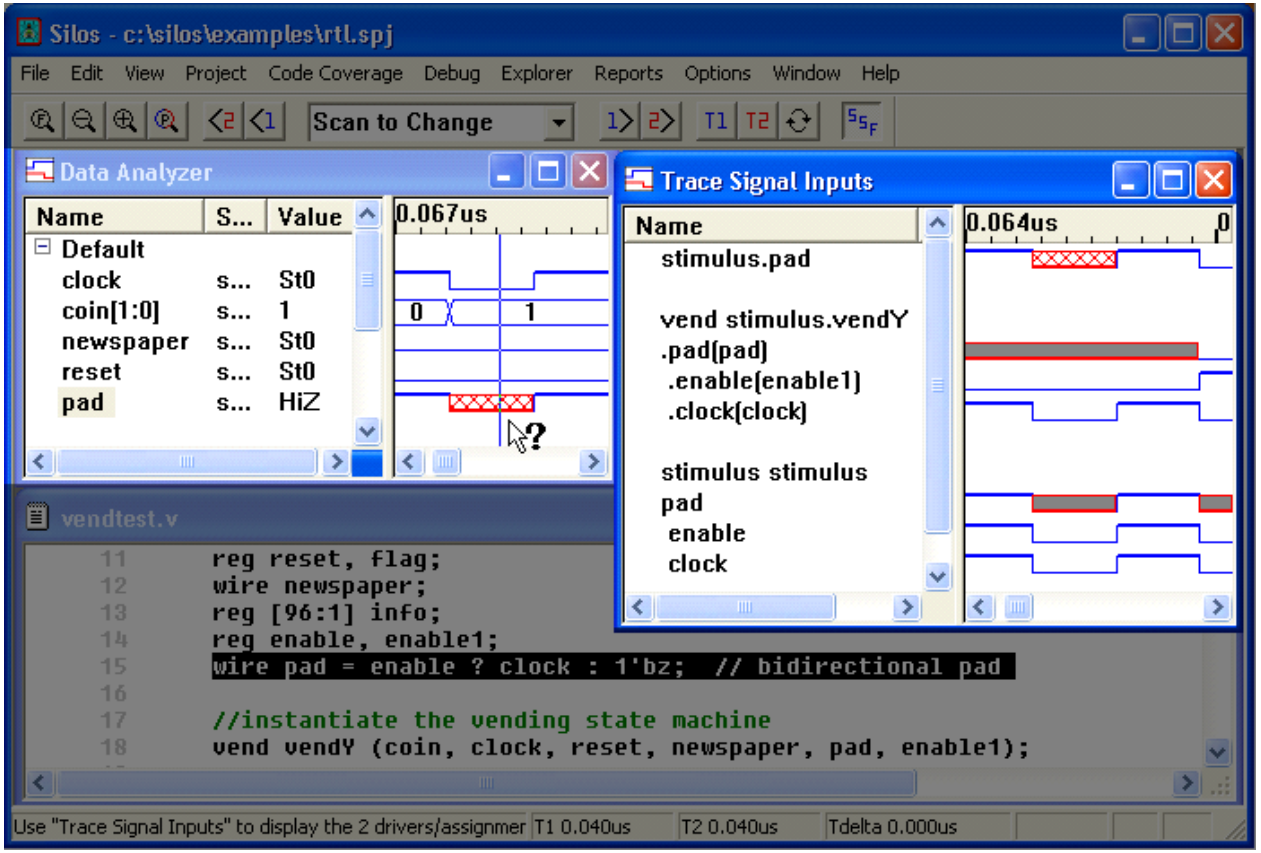
Visual Debug - “pad”



To find the cause of the Unknown level on “pad”, right mouse click on the signal named “pad”.

SILVACO

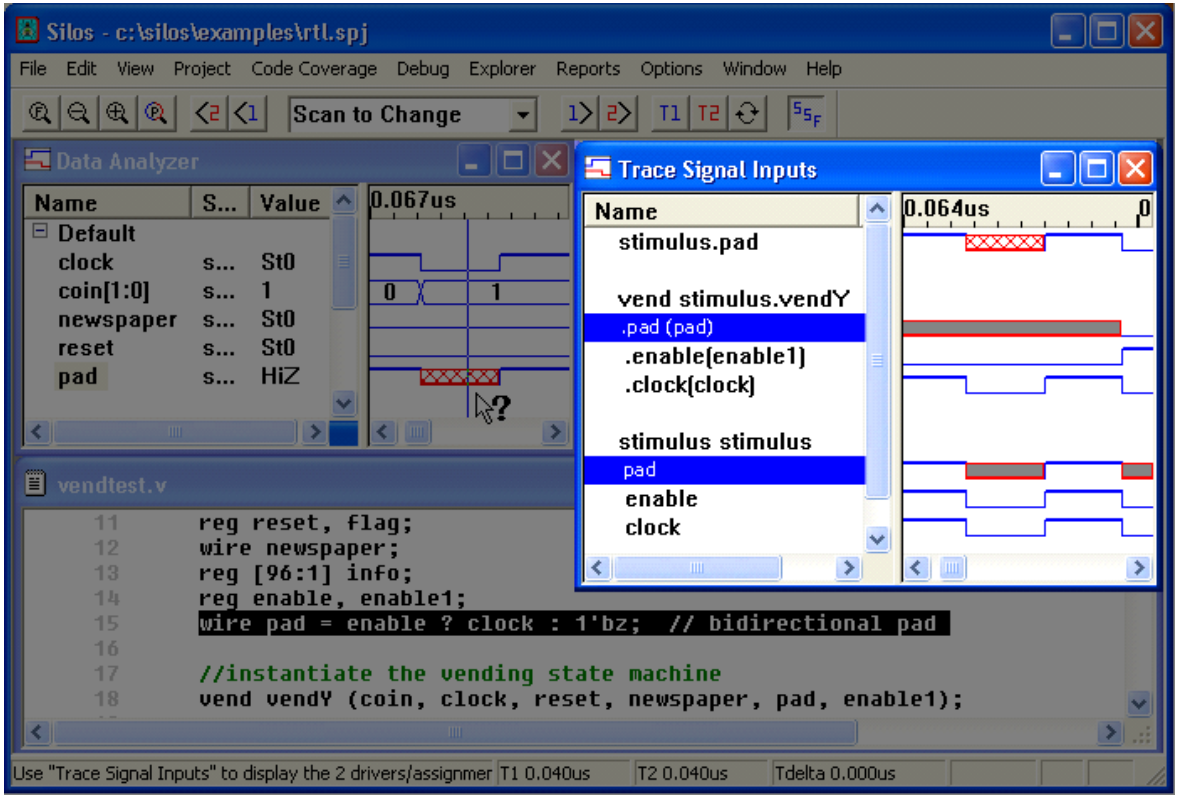
Visual Debug - “pad”



..and choose Trace Signal Inputs from the context menu.

SILVACO

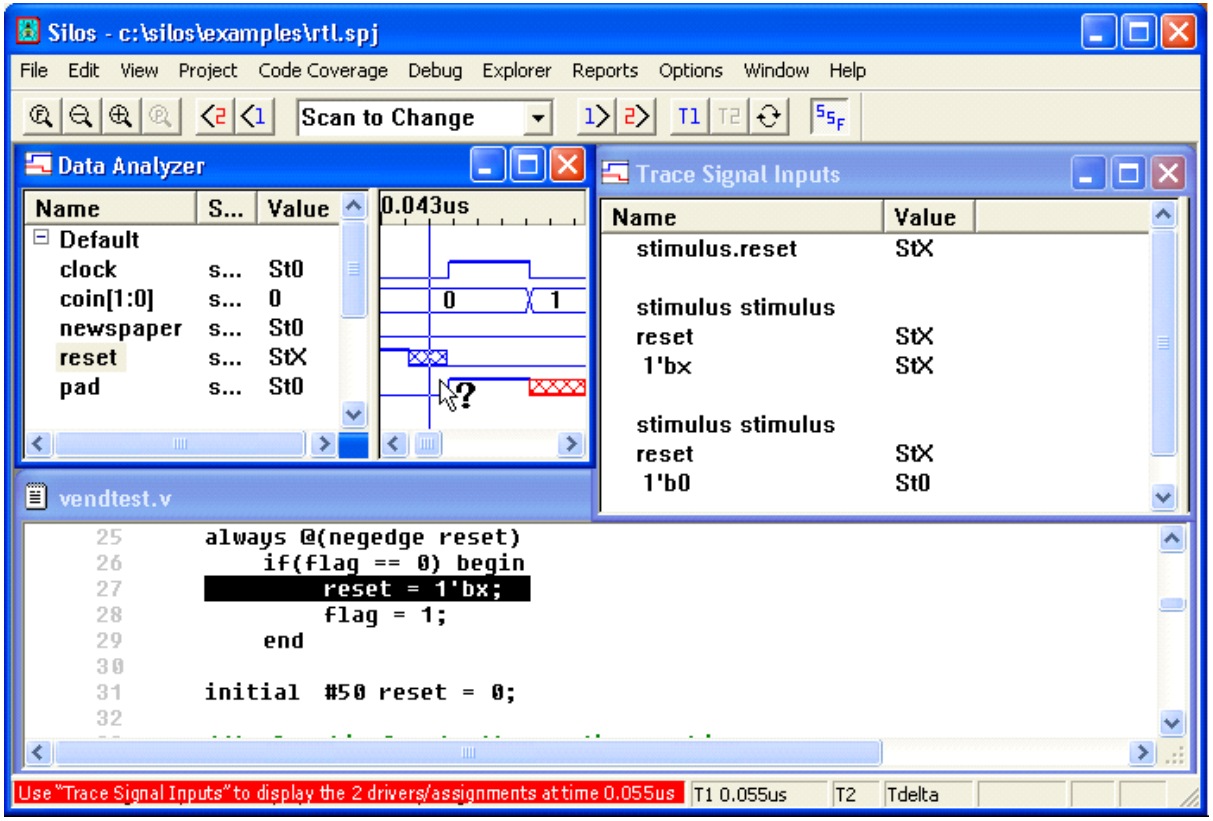
Visual Debug - “stimulus.pad”



Unknown level on “stimulus.pad” is caused by both enables being low. Left mouse clicking on either driver name “pad” will highlight the corresponding line of source for that driver.

SILVACO

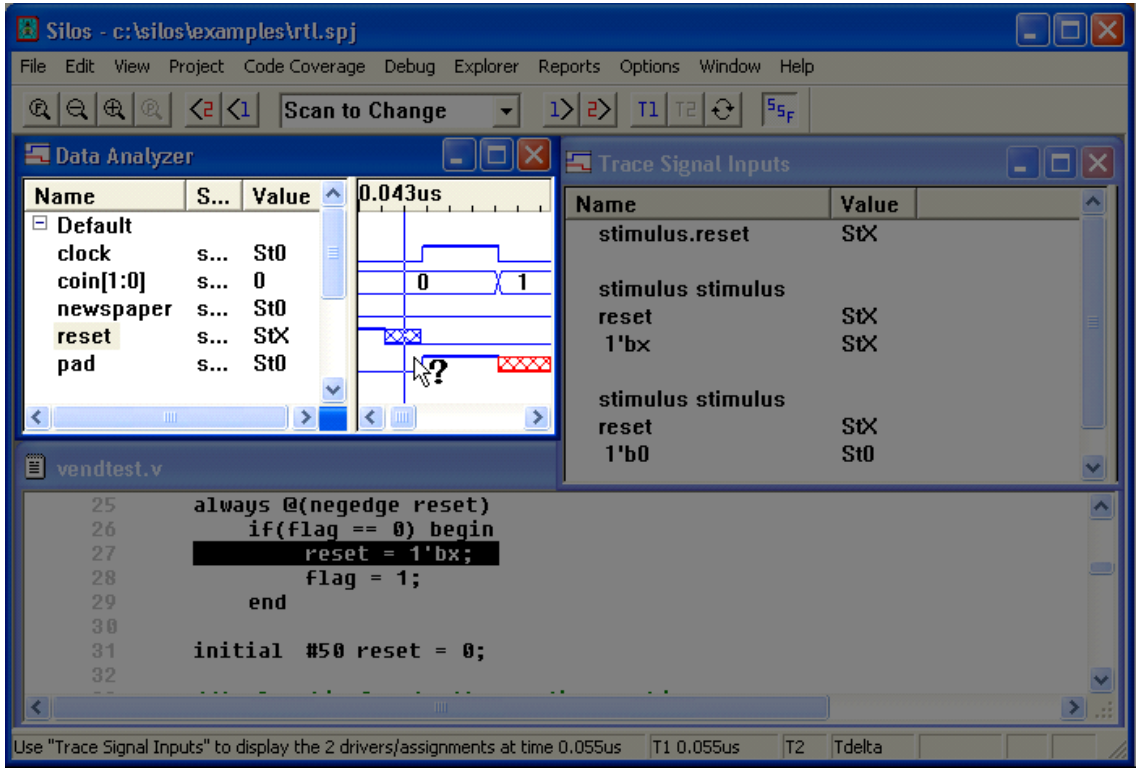
Visual Debug - “stimulus.pad”



Red flashing message in status window warns user of multiple drivers or assignments at a time point while “Show Source” is on.

SILVACO

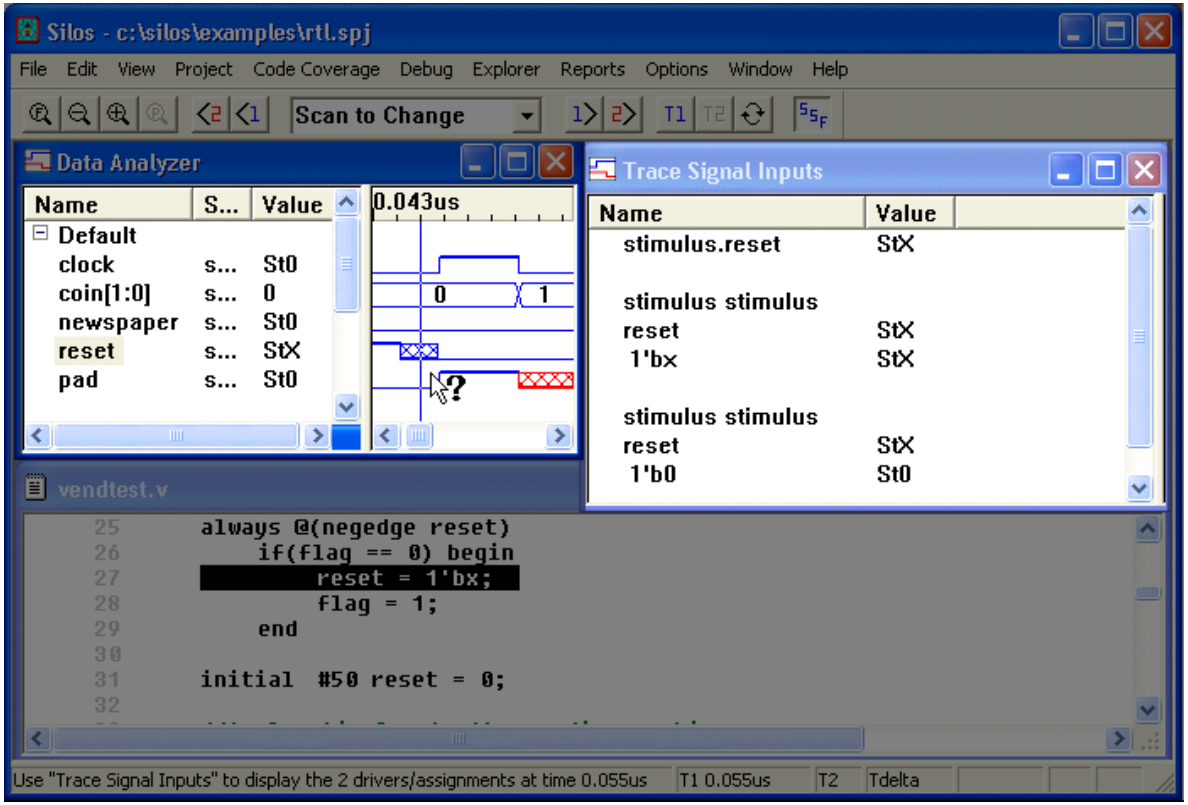
Visual Debug - “reset”



- To find the cause of the Unknown level on register “reset”, right mouse click on the signal named “reset”…

SILVACO

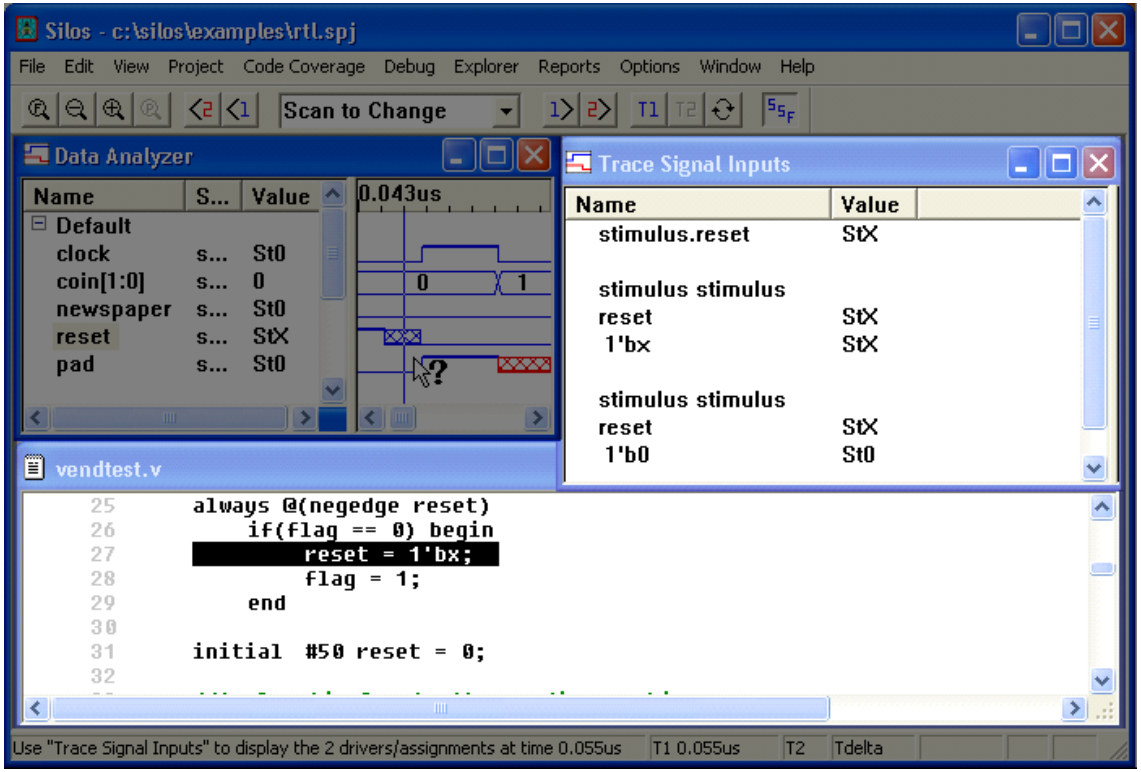
Visual Debug - “reset”



...and choose Trace Signal Inputs from the context menu.

SILVACO

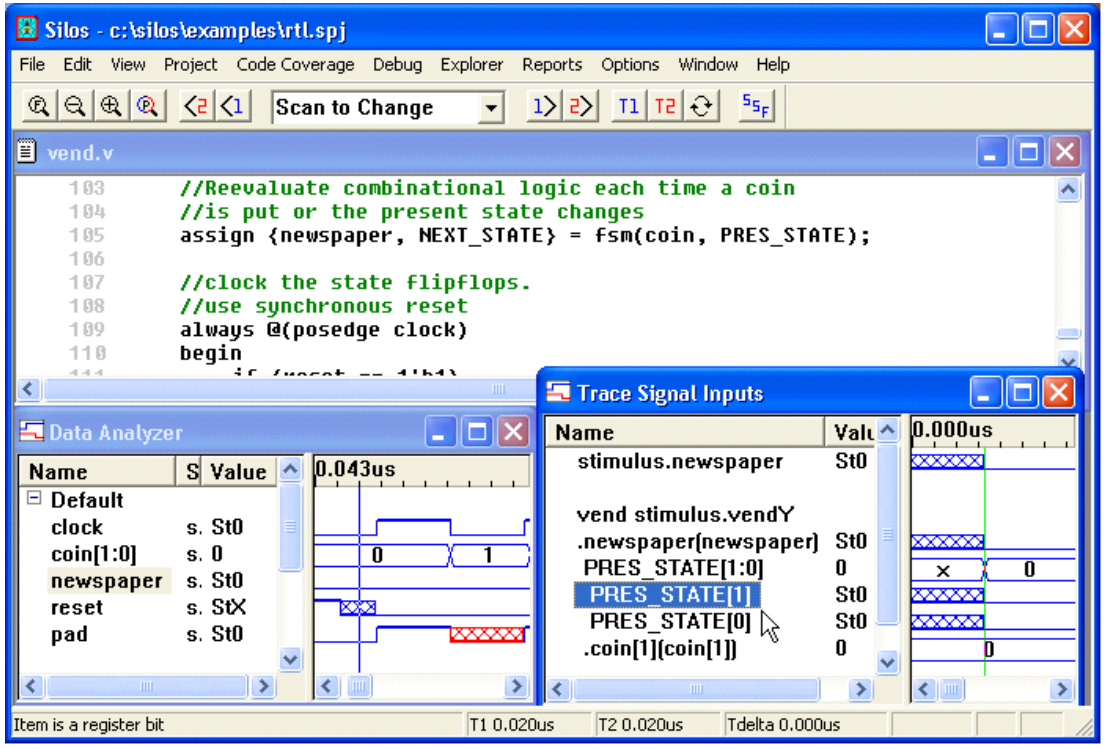
Visual Debug - “reset”



The Trace Signal Inputs window shows that two statements assigned to register “stimulus.reset” at T_1=.0.055us. The first assignment listed in the Trace Signal Inputs window is the last assignment to ‘reset’ that executed at this time point. The second statement listed is the next to last assignment to “reset” that executed. Left mouse clicking on the first statement for “reset” shows line 27 of file “vendtest.v” is the last line executed.

SILVACO

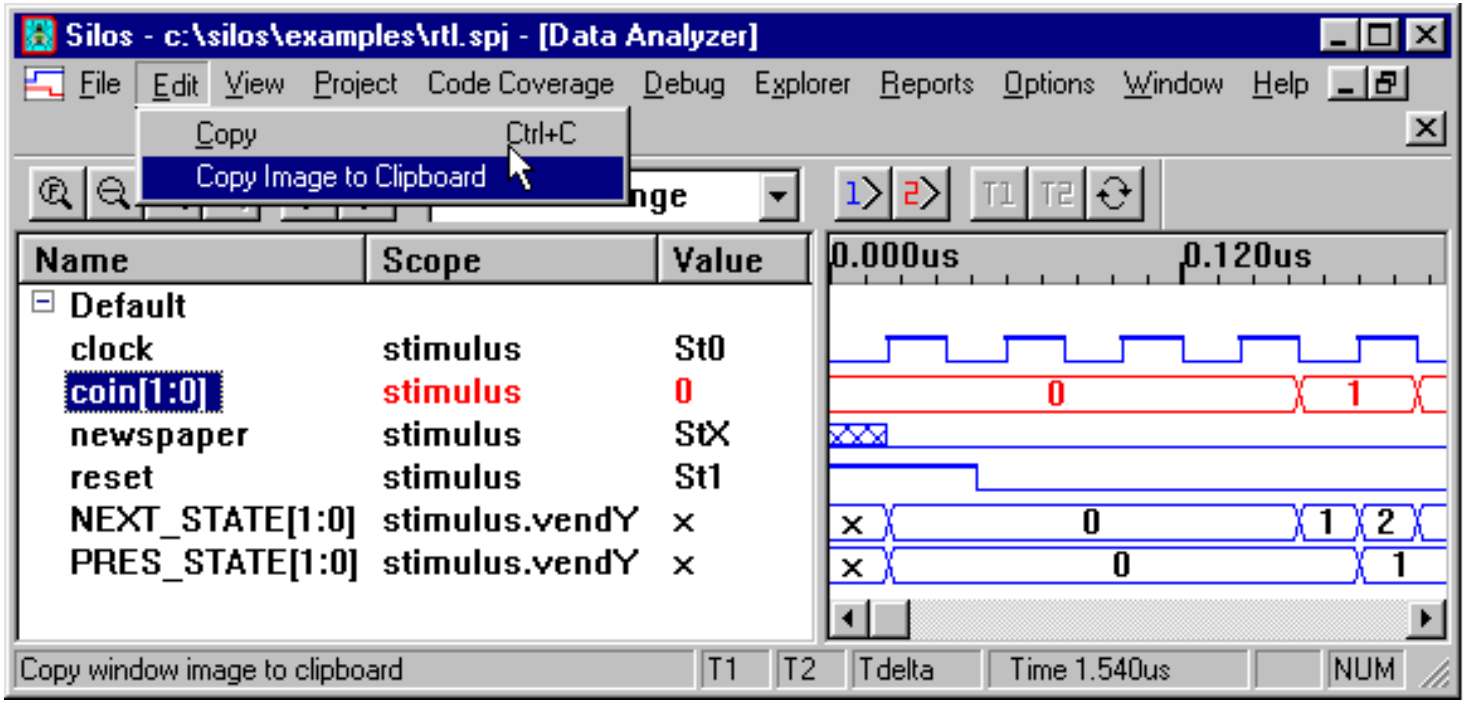
Visual Debug



Tracing on “newspaper” in the Data Analyzer brings up the Trace Signal Inputs window, which shows “PRES_STATE[1:0]” and “coin[1]” are the right hand side (rhs) variables that assign the value for newspaper. Double clicking on “PRES_STATE[1]” will refresh the Trace Signal Inputs window, put “PRES_STATE[1]” at the top, and display the rhs variables for “PRES_STATE[1]”.

SILVACO

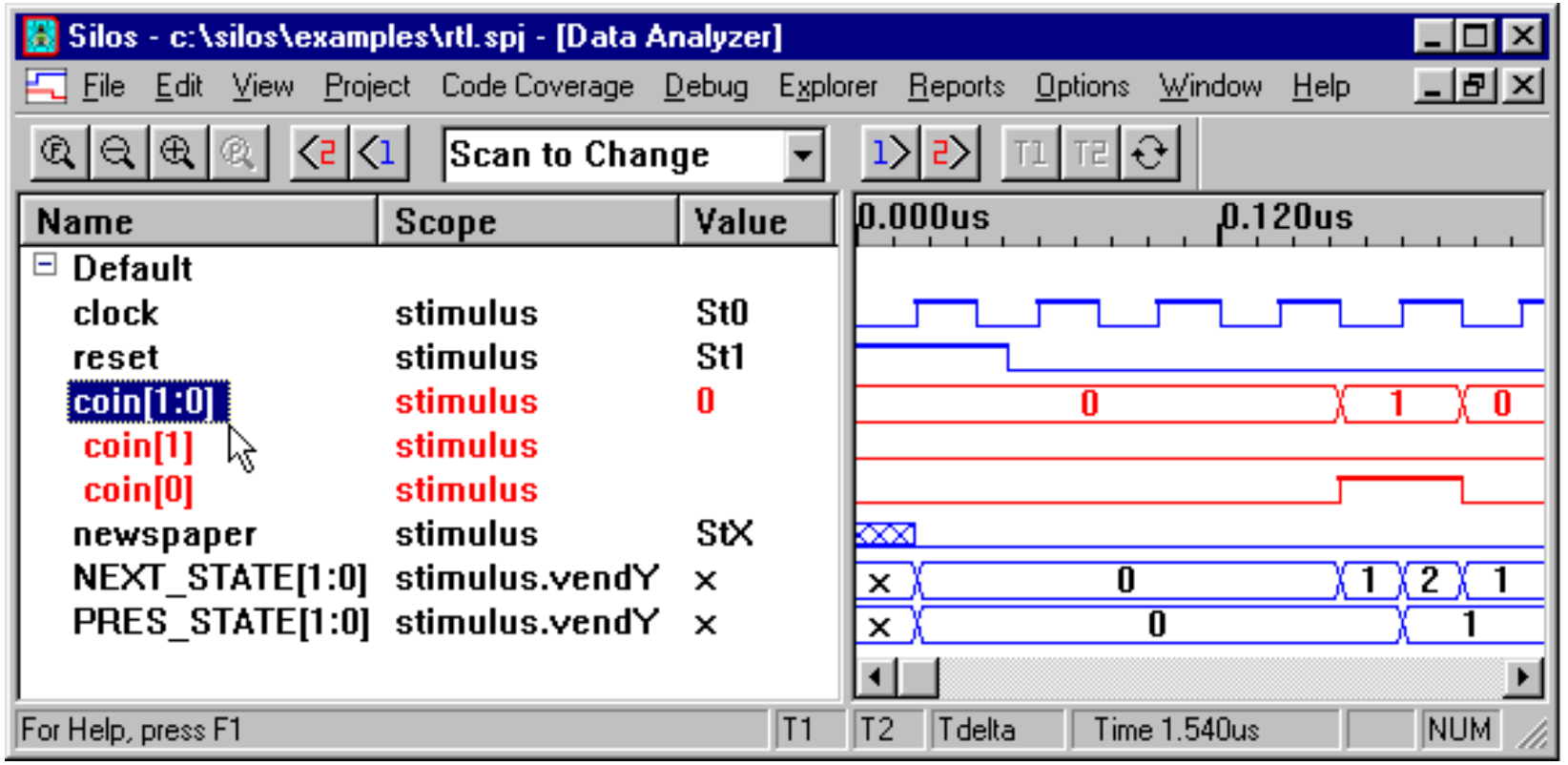
Report Generation



Select the “Edit/Copy Image to Clipboard” menu to copy the Data Analyzer so it can be pasted into MS Word. You can use the “Edit/Copy” menu, or “Ctrl+C” on the keyboard to copy the full path name for a signal to the clipboard.

SILVACO

Vector Display



Double click on vector “coin[1:0] to expand and hide the bits.

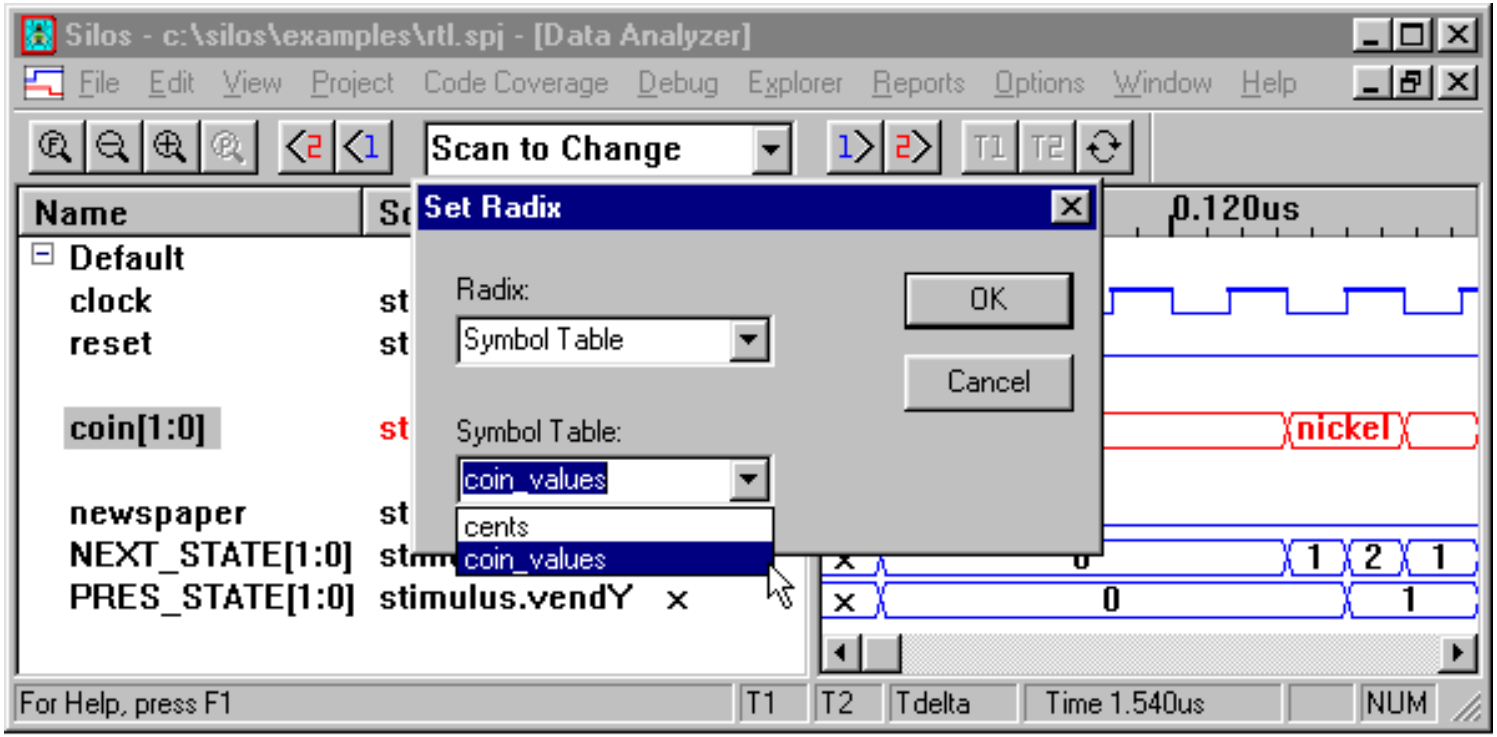
SILVACO

Agenda: State Machine Design Entry

- Introduction
- Starting Silos Project
- Explorer and Analyzer
- Source Code Debugging
- State Machine Design Entry
- Advanced Debugging Features
- Finite State Machine Example
- Gate Level Debugging

SILVACO

State Machine Values

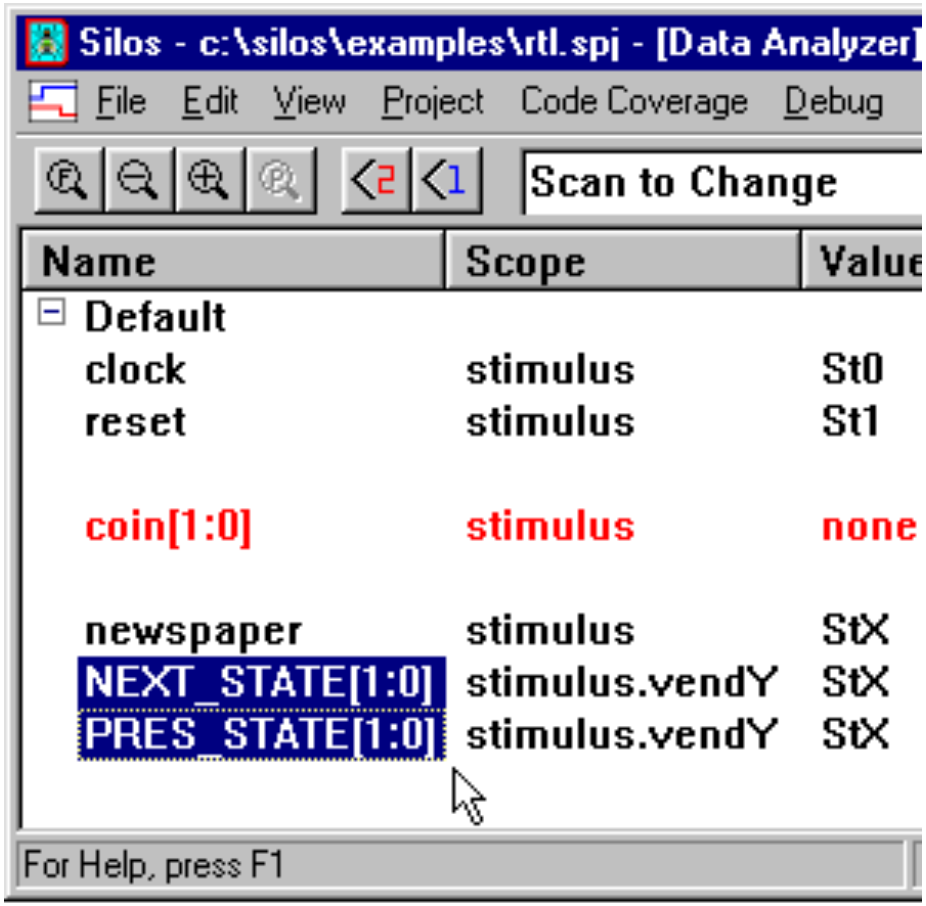


Select “coin[1:0]”.
Right mouse click to see context menu. Then select “Set Radix”.

Use the drop down arrows to display Symbol Table and coin_values.

SILVACO

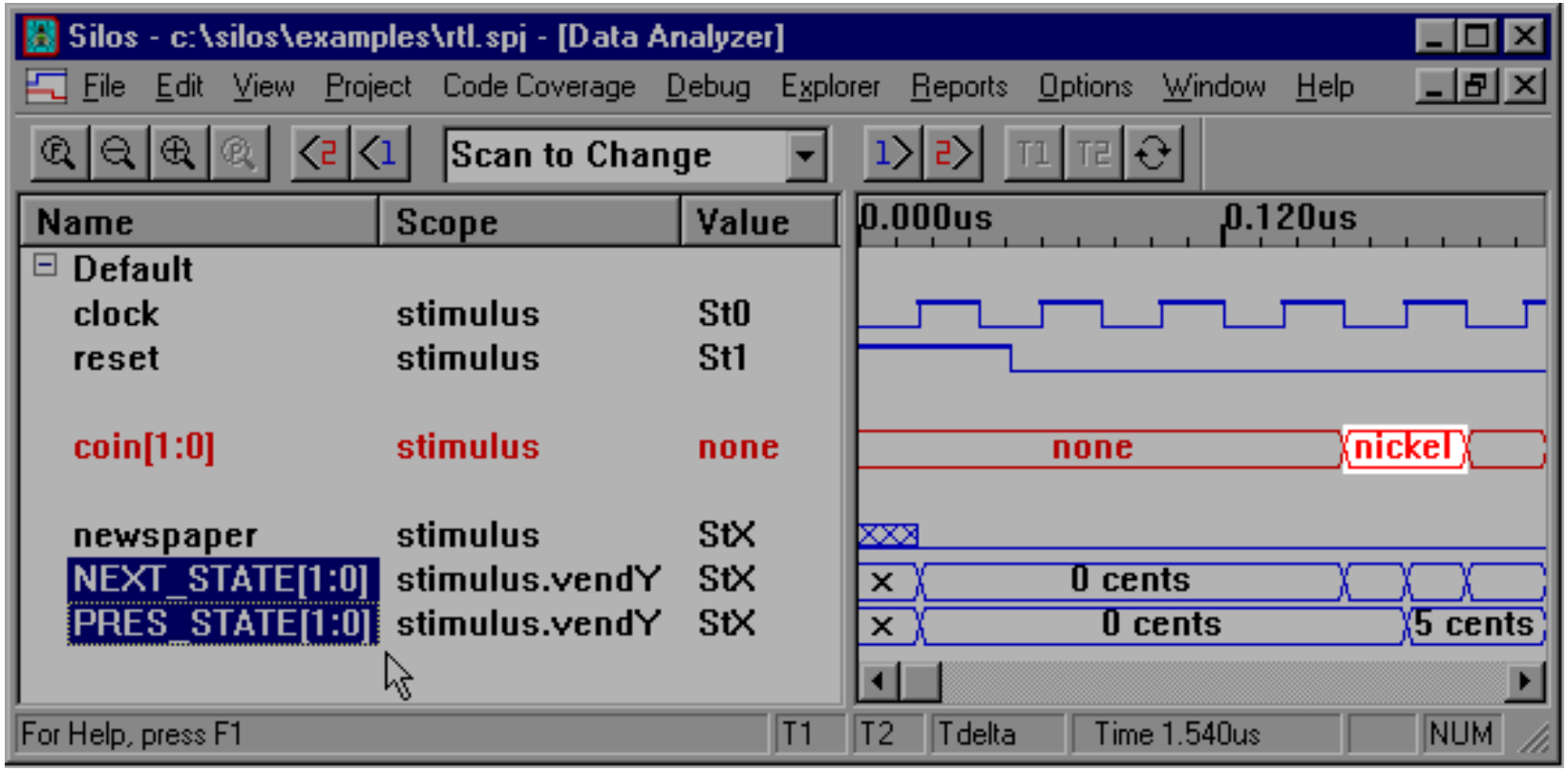
State Machine Values



Make sure you click on the minus sign ⊟ to save the new radixes.

SILVACO

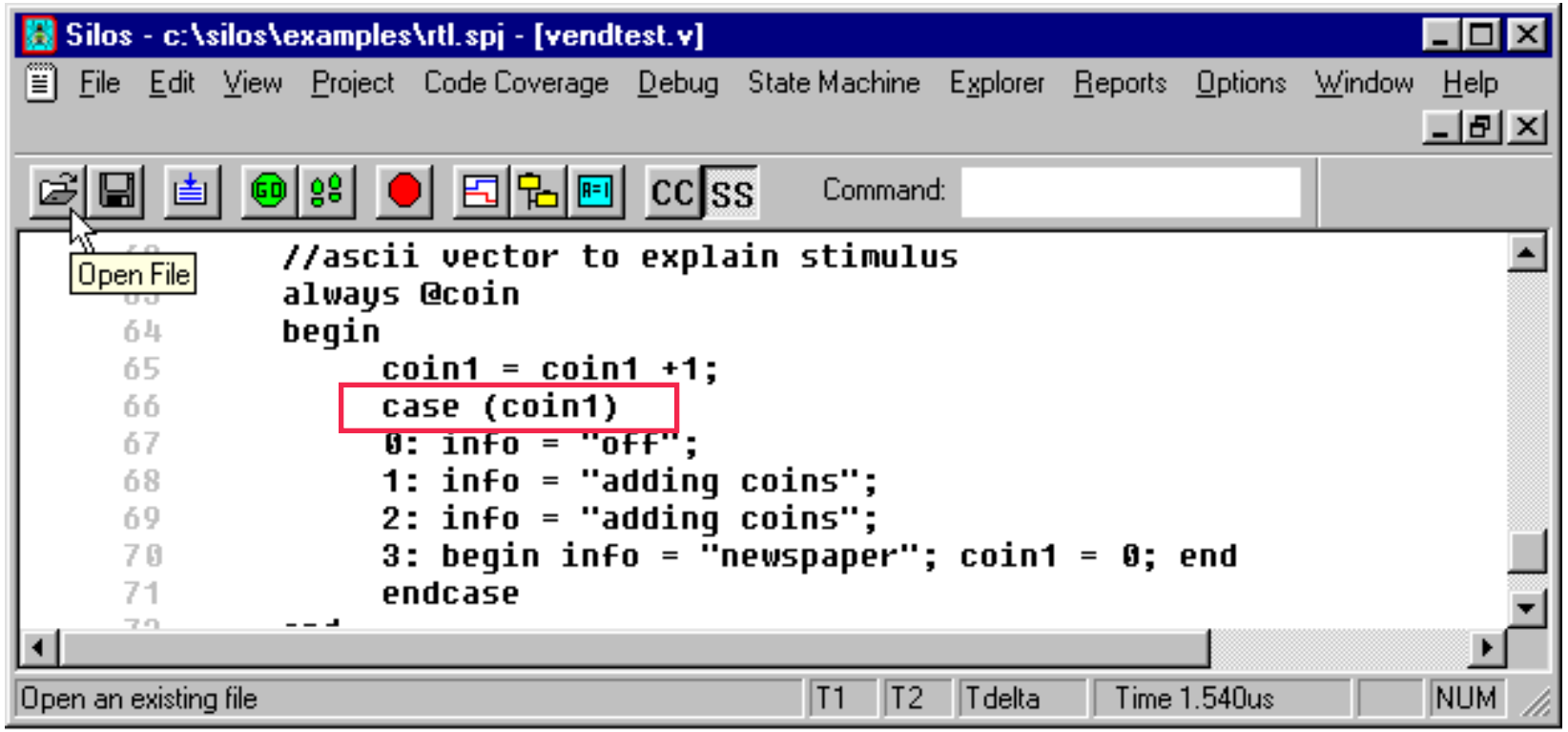
State Machine Values



Symbolic names for state values for vectors.

SILVACO

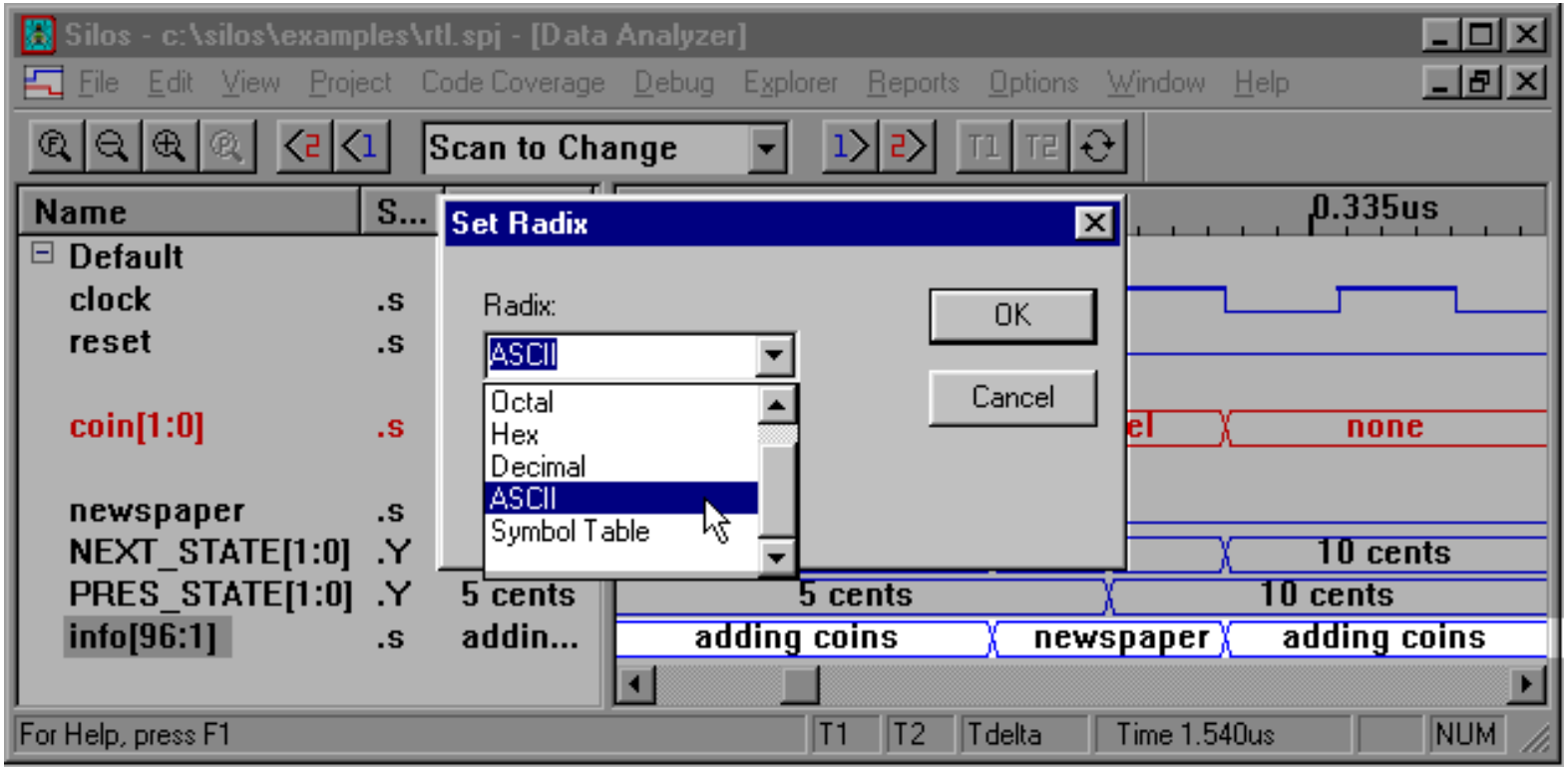
Waveform Annotation



Case statement for ASCII vector “info”.

SILVACO

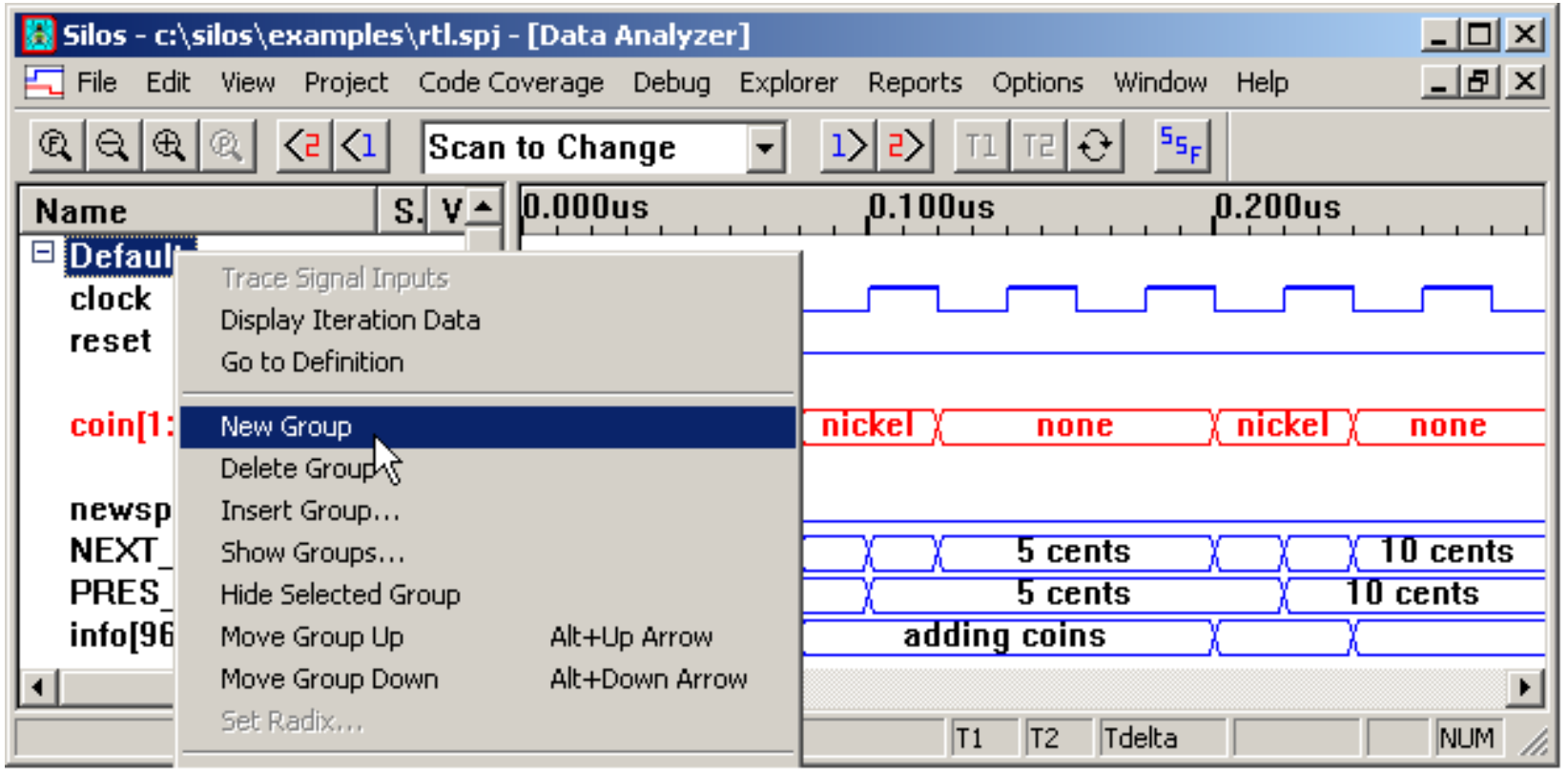
Waveform Annotation



ASCII text describes the simulation.

SILVACO

Grouping Waveforms

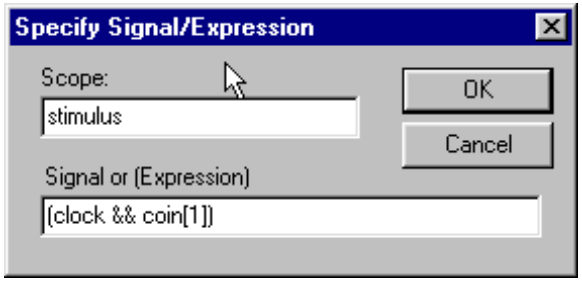


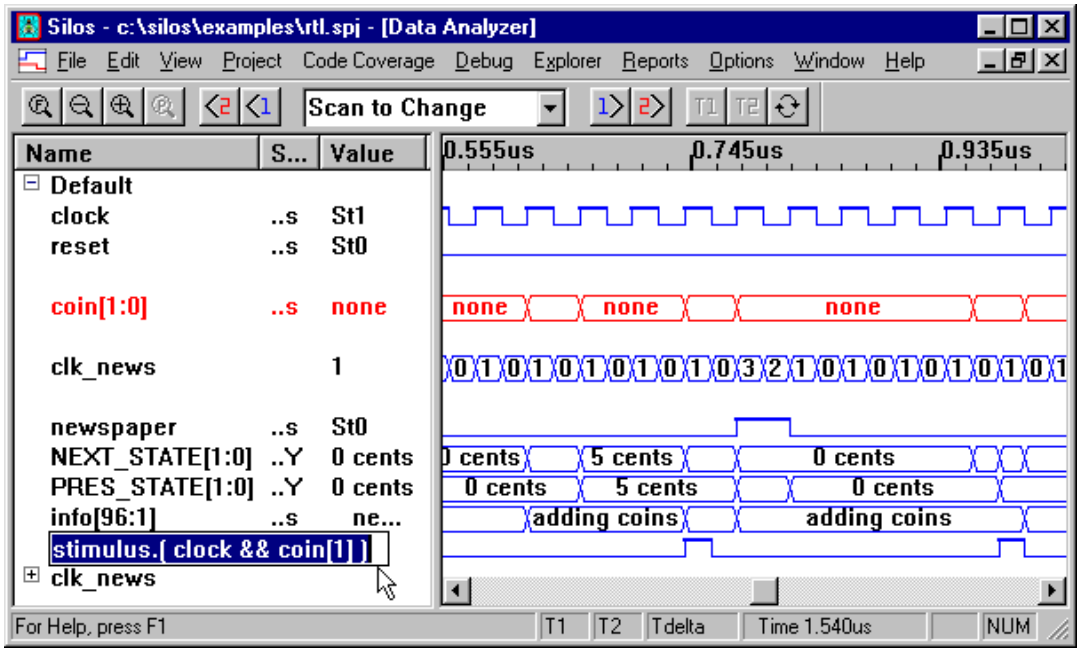
Click with right mouse button in Signal list Box

Selections for groups.

SILVACO

Conditional Search



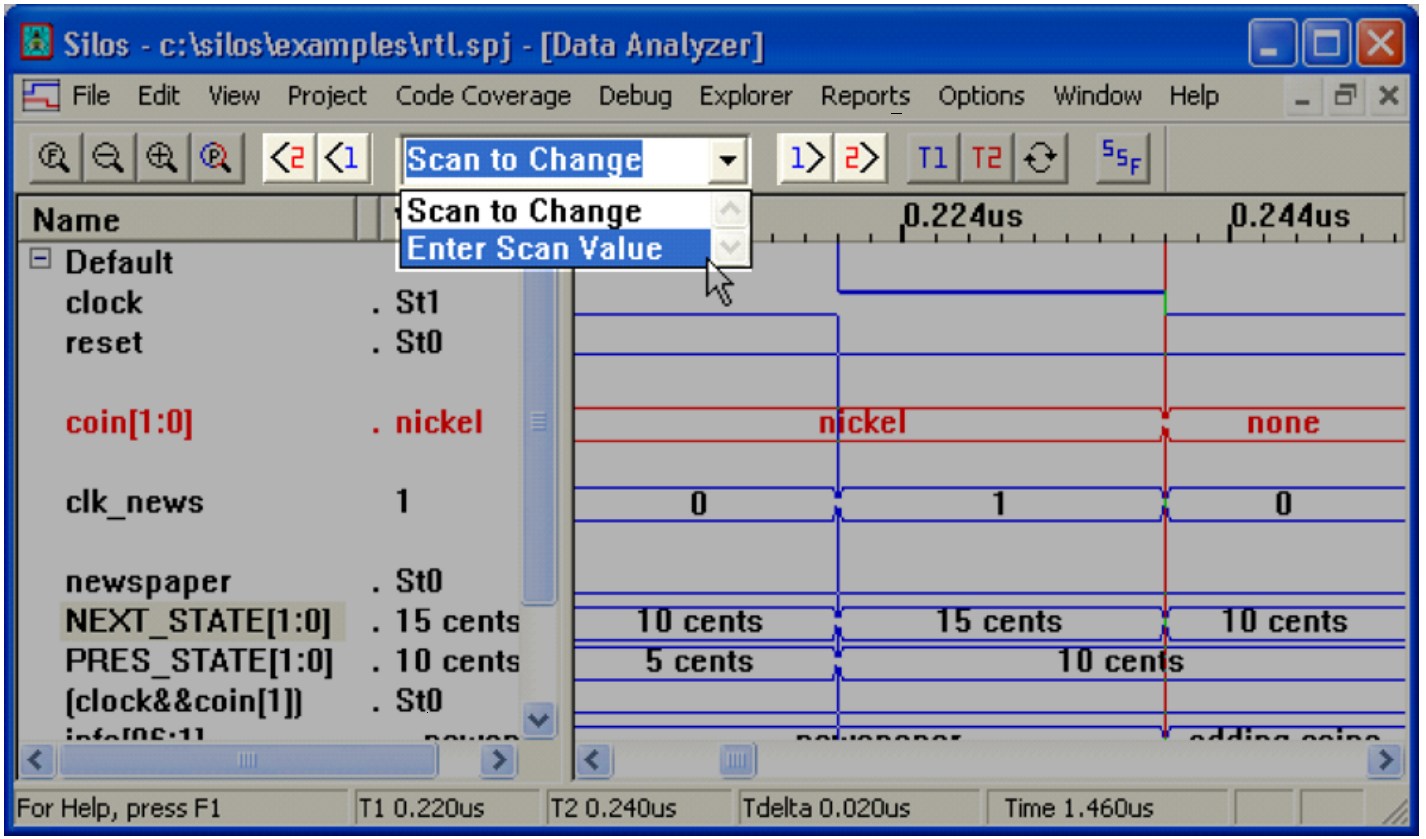


Expression “(clock && coin[1])”.
Notice that if you click on the expression a second time, you can modify the expression.

SILVACO

Data Analyzer - Scan Buttons

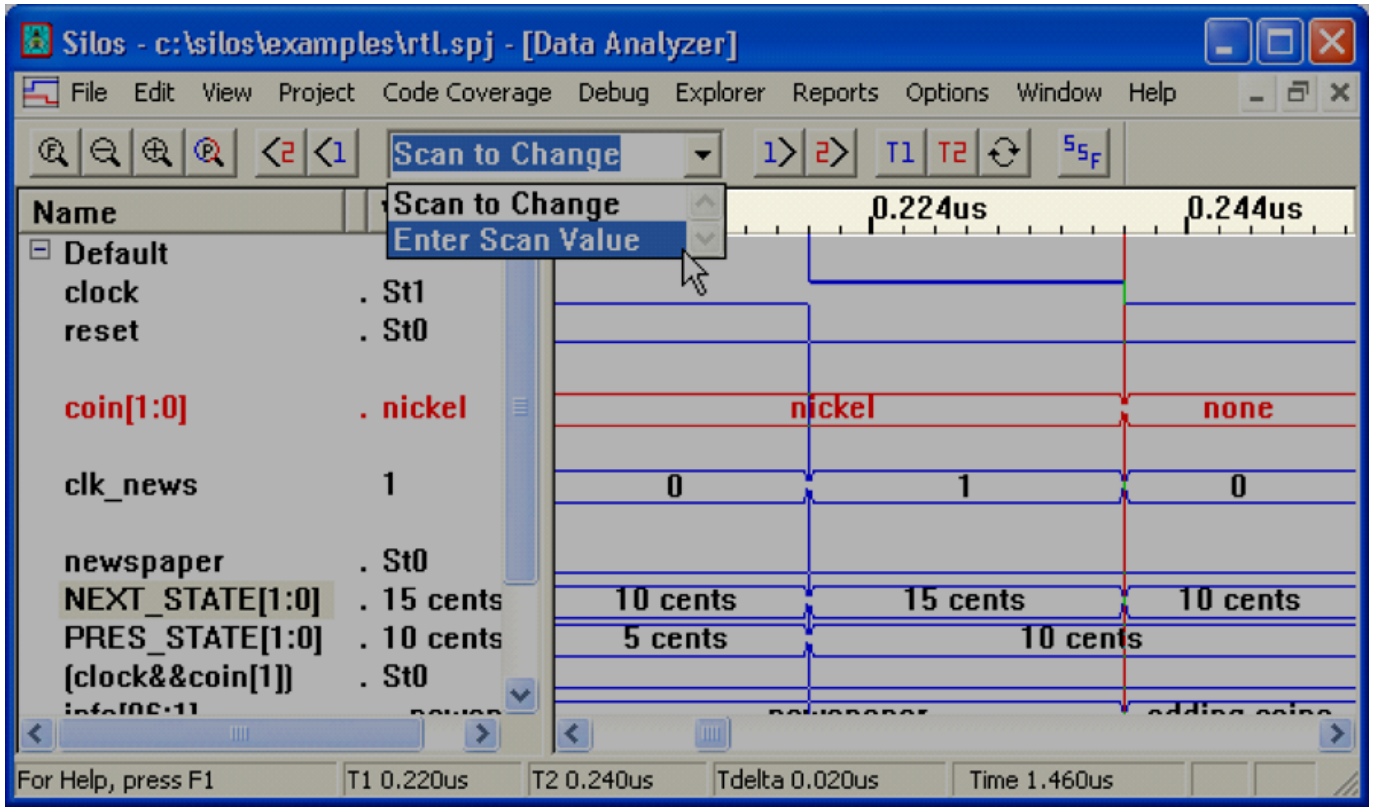
Scan left buttons Scan Value Scan right buttons



SILVACO

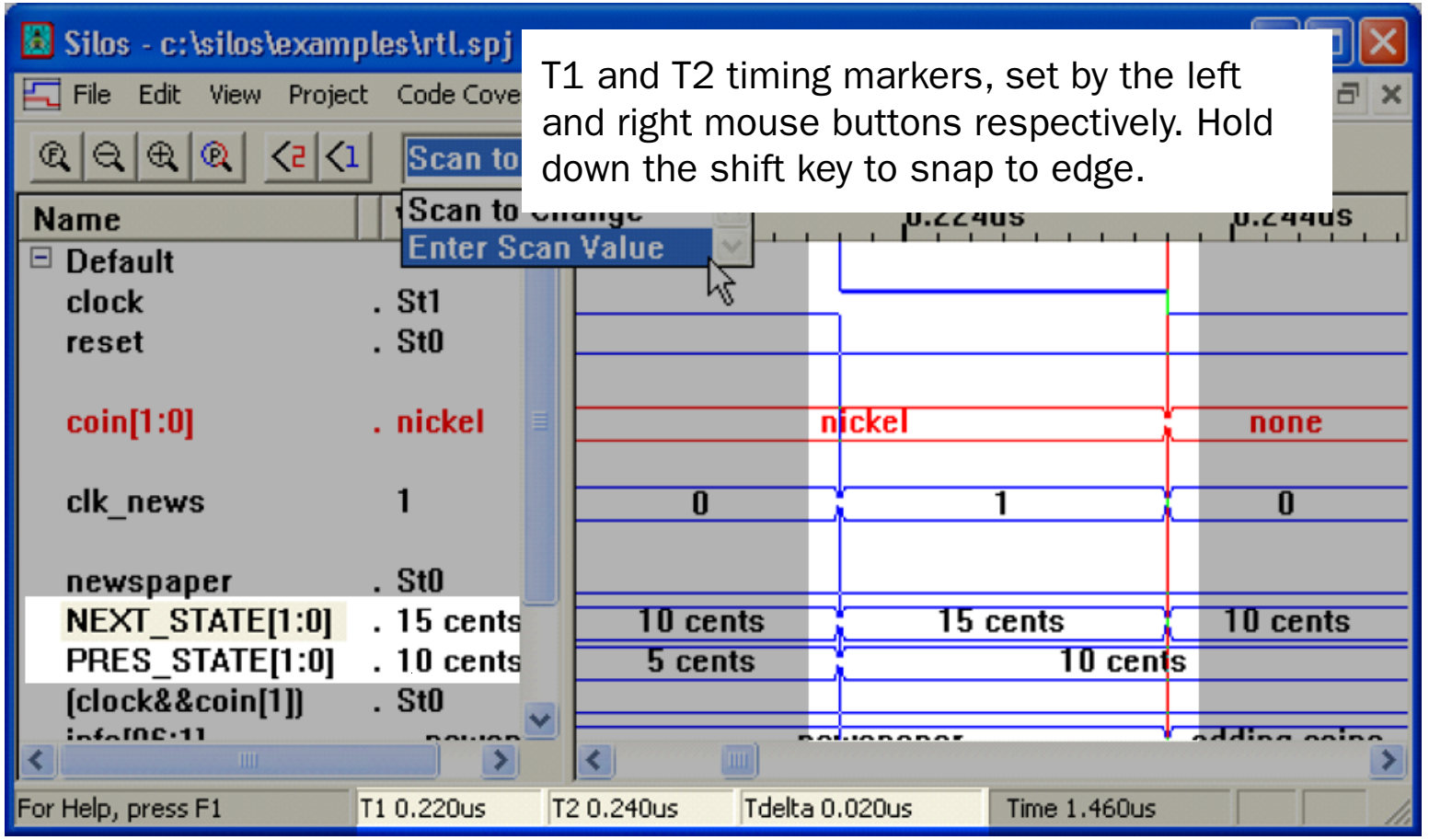
Data Analyzer - Timescale

Put the cursor here to display the timescale, T1 and T2 times, and the delta time.



SILVACO

Data Analyzer



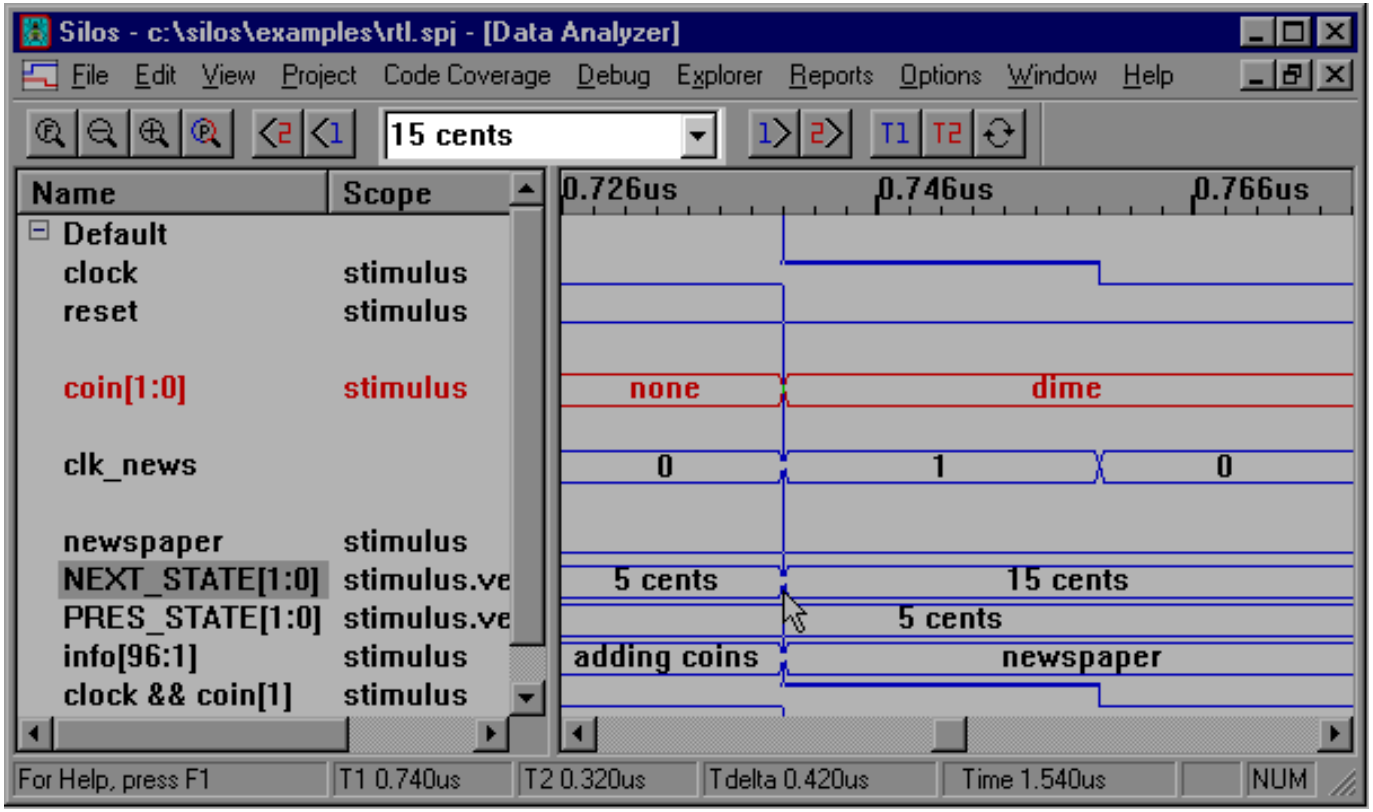
T1 and T2 timing markers, set by the left and right mouse buttons respectively. Hold down the shift key to snap to edge.

Value for the signal

T1, T2, and delta time

SILVACO

Data Analyzer - Scan to Value



Scan to value does a character search, so you can scan a signal that is a single bit, a vector, or that uses symbolic names.

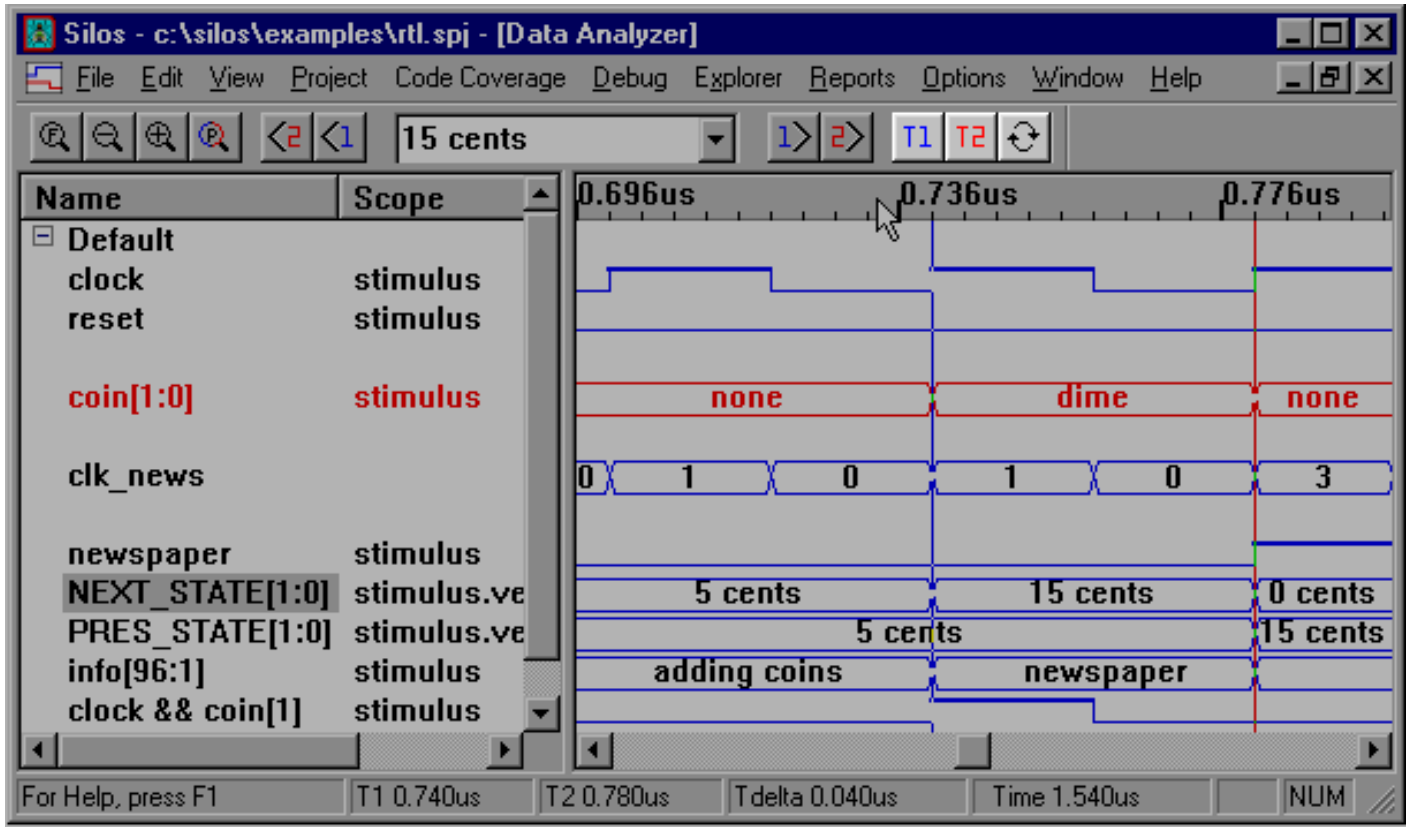
SILVACO

Data Analyzer - Pan Buttons

Pan to T1 button.

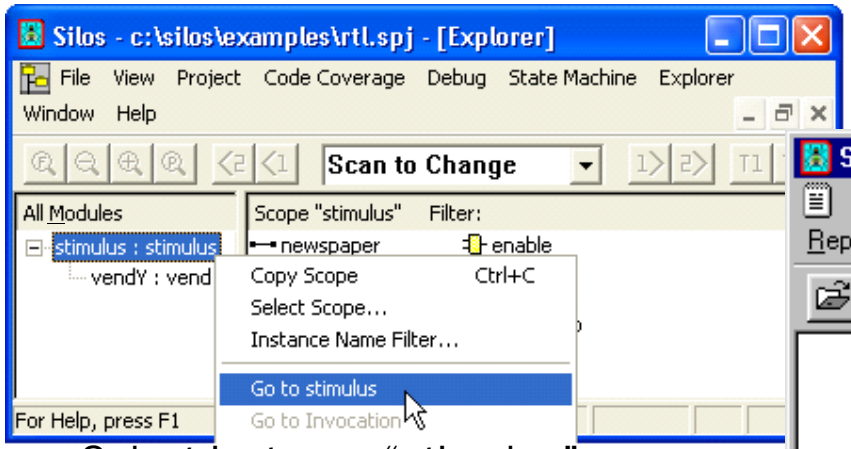
Pan to T2 button.

Pan Last.

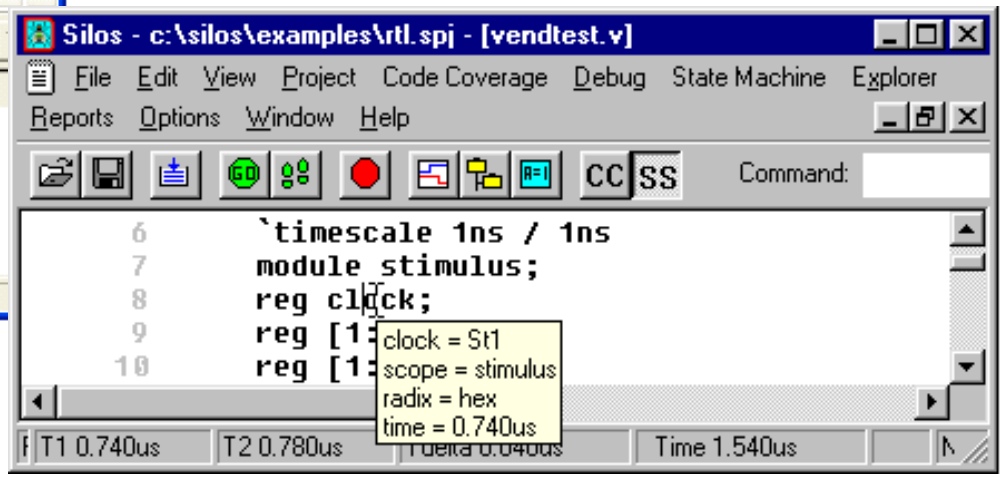


SILVACO

Data Tips



Select instance “stimulus”, open the context menu with the right mouse button, and then select “Go to stimulus”.



Data Tips display value, scope, radix, and simulation time point for variable clock. The time point can be set by the “T1” timing marker in the Data Analyzer. The Data Tips can be turned on/off and the radix can be changed by using the context menu in the source window.

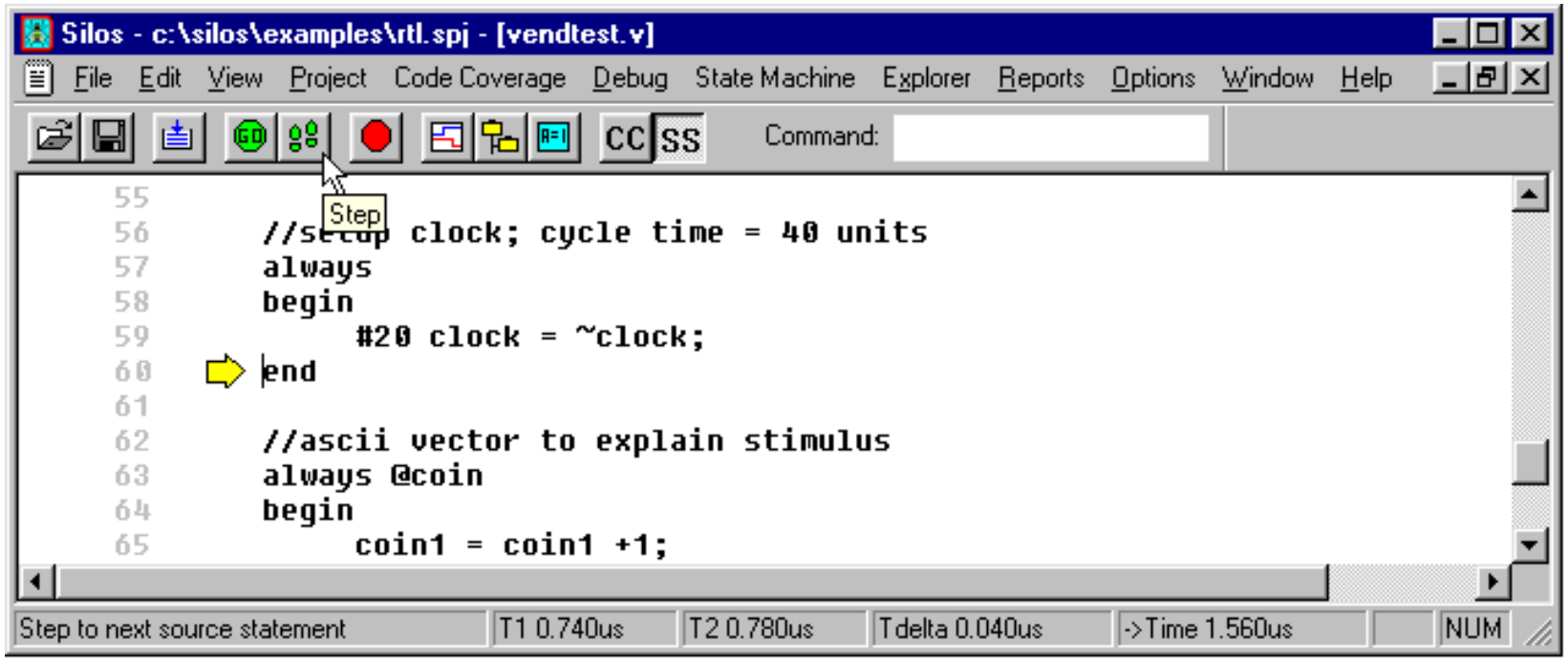
SILVACO

Agenda: Advanced Debugging Features

- Introduction
- Starting Silos Project
- Explorer and Analyzer
- Source Code Debugging
- State Machine Design Entry
- Advanced Debugging Features
- Finite State Machine Example
- Gate Level Debugging

SILVACO

Single Stepping

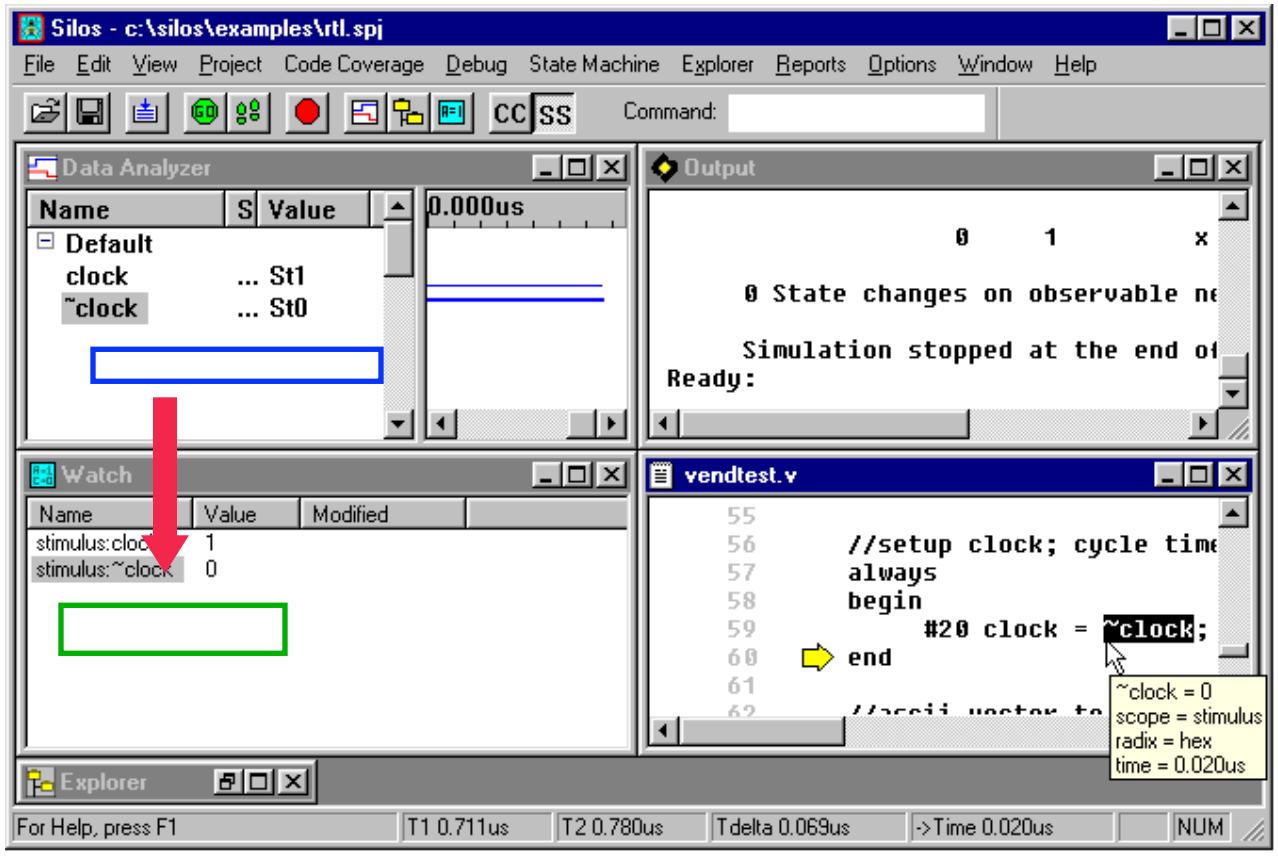


Click on the “Step” button to open the source window to the line that is currently executing.

SILVACO

Single Stepping

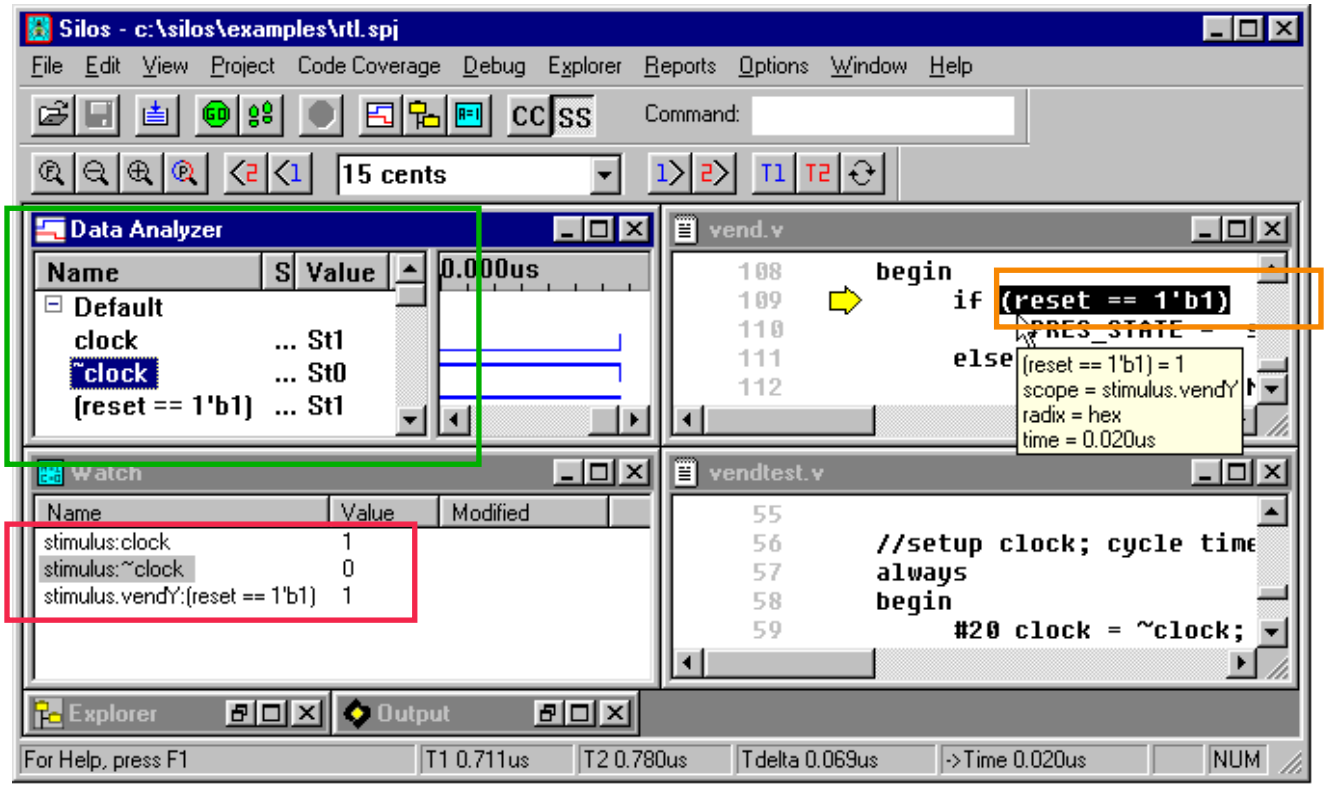
Drag and drop expression “~clock” from file “vendtest.v” [1] to the Data Analyzer and Watch Window [2].



Notice the Data Tip shows the instantenous value for expression “~clock” is “0” at the current timestep. However, the Data Analyzer is not updated until the end of the timestep, so its value for “~clock” is “1”.

SILVACO

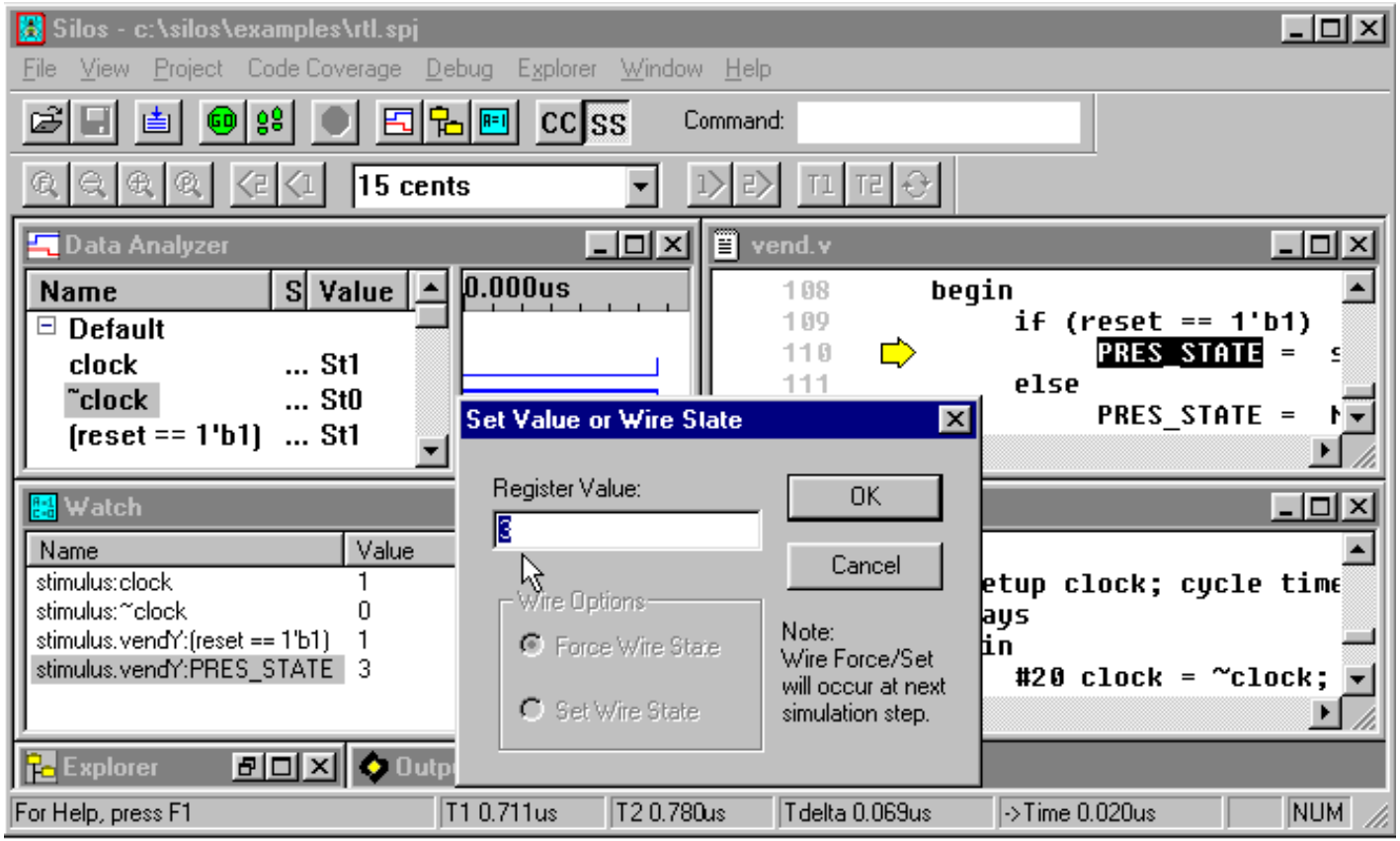
Single Stepping



Drag and drop expression “reset == 1’b1” [1] into the Data Analyzer [2] and Watch Windows [3].

SILVACO

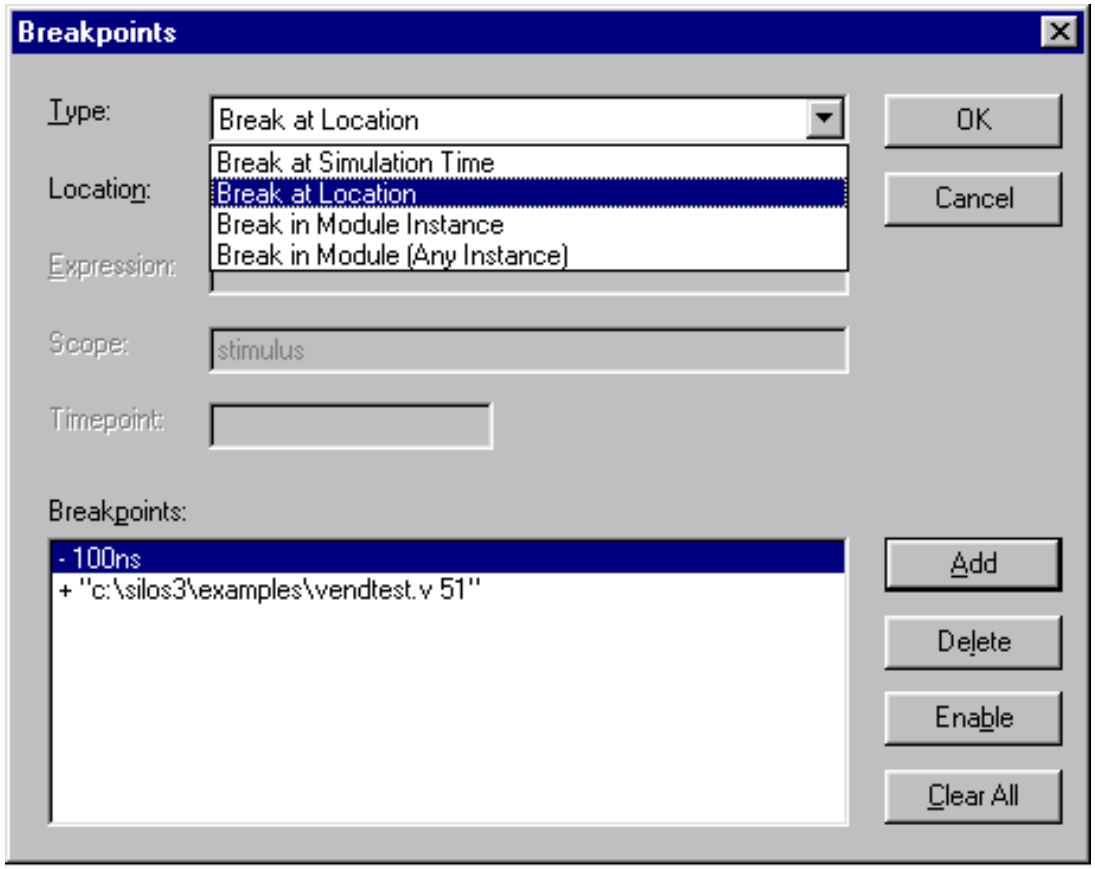
Set and Force



Set the value for register PRES_STATE to “3”.

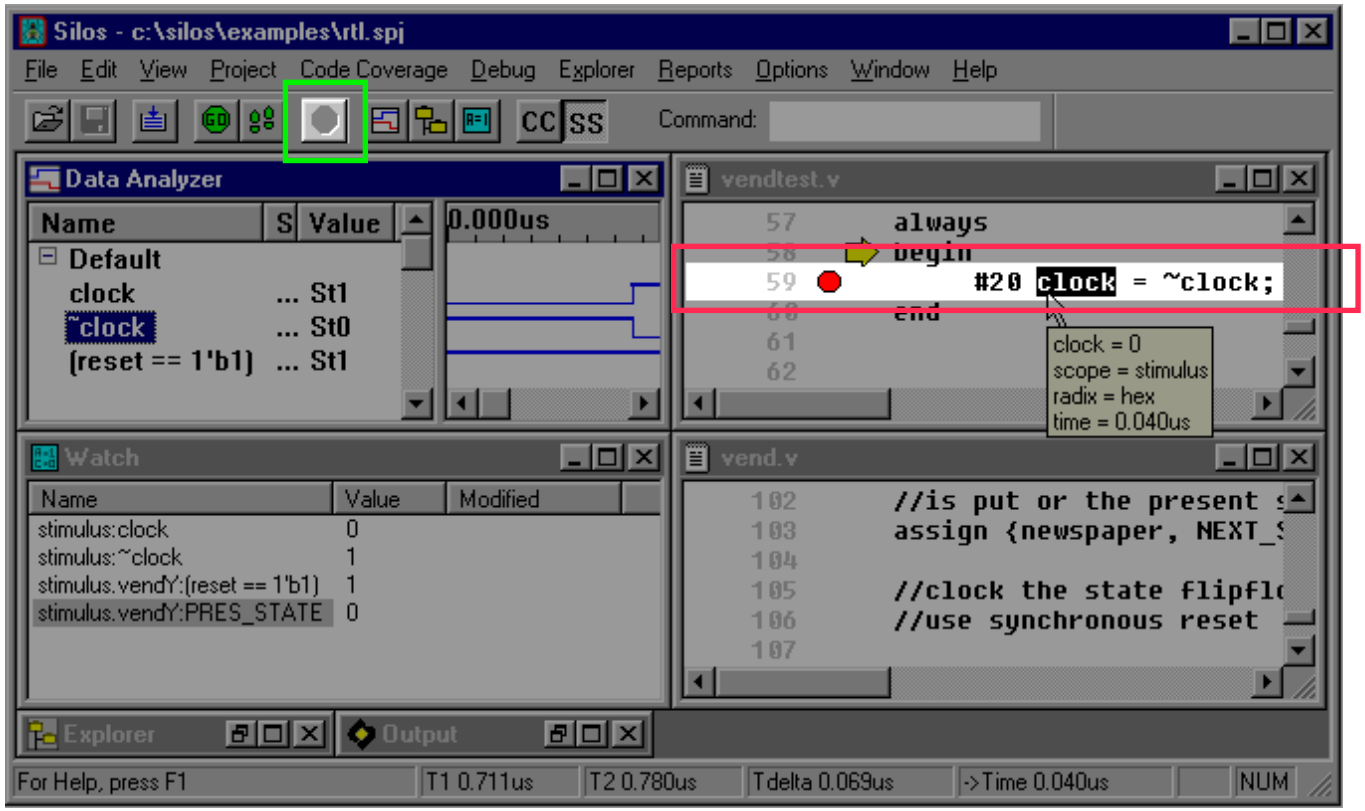
SILVACO

Breakpoints



SILVACO

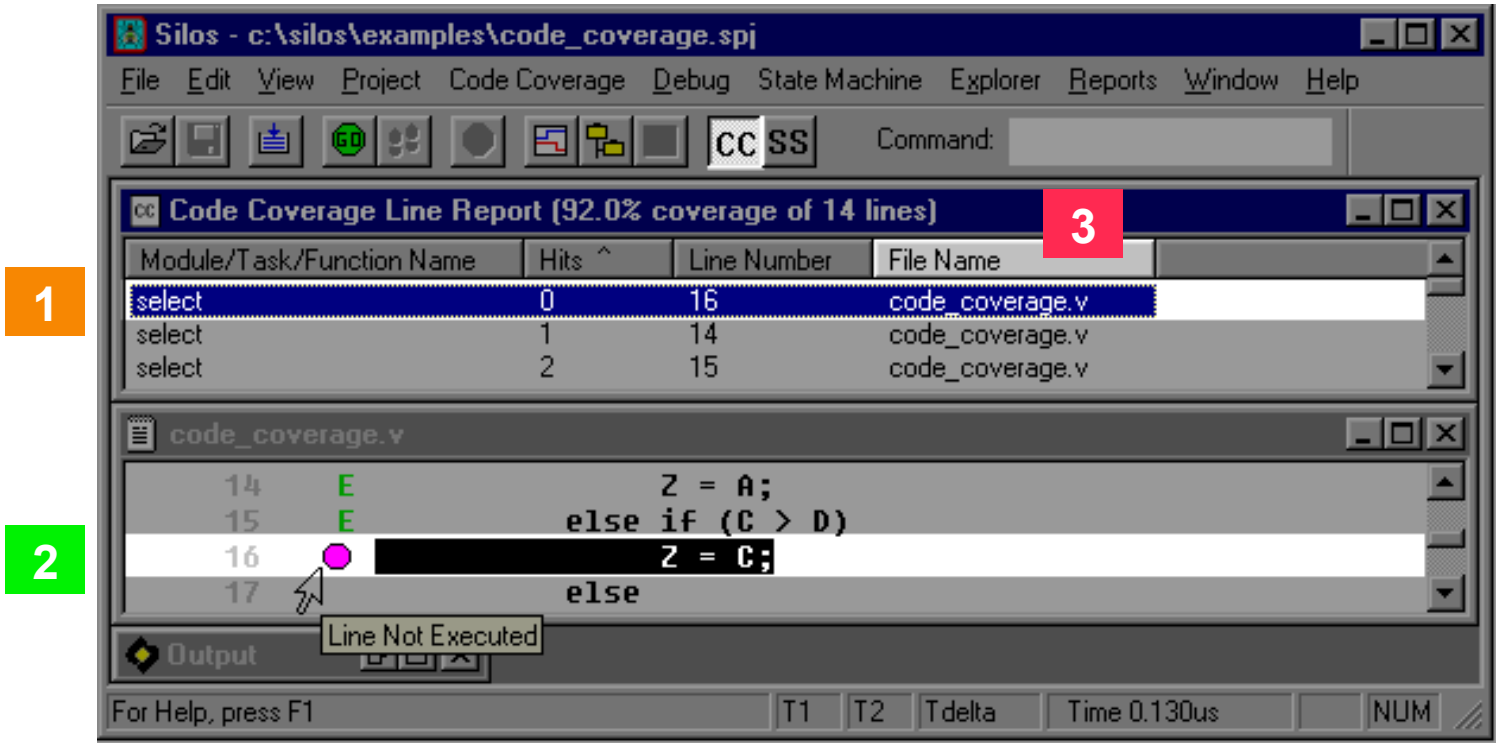
Breakpoints



Put the cursor on this line and click on the Toggle Breakpoint button [1] to set the red breakpoint stop sign [2].

SILVACO

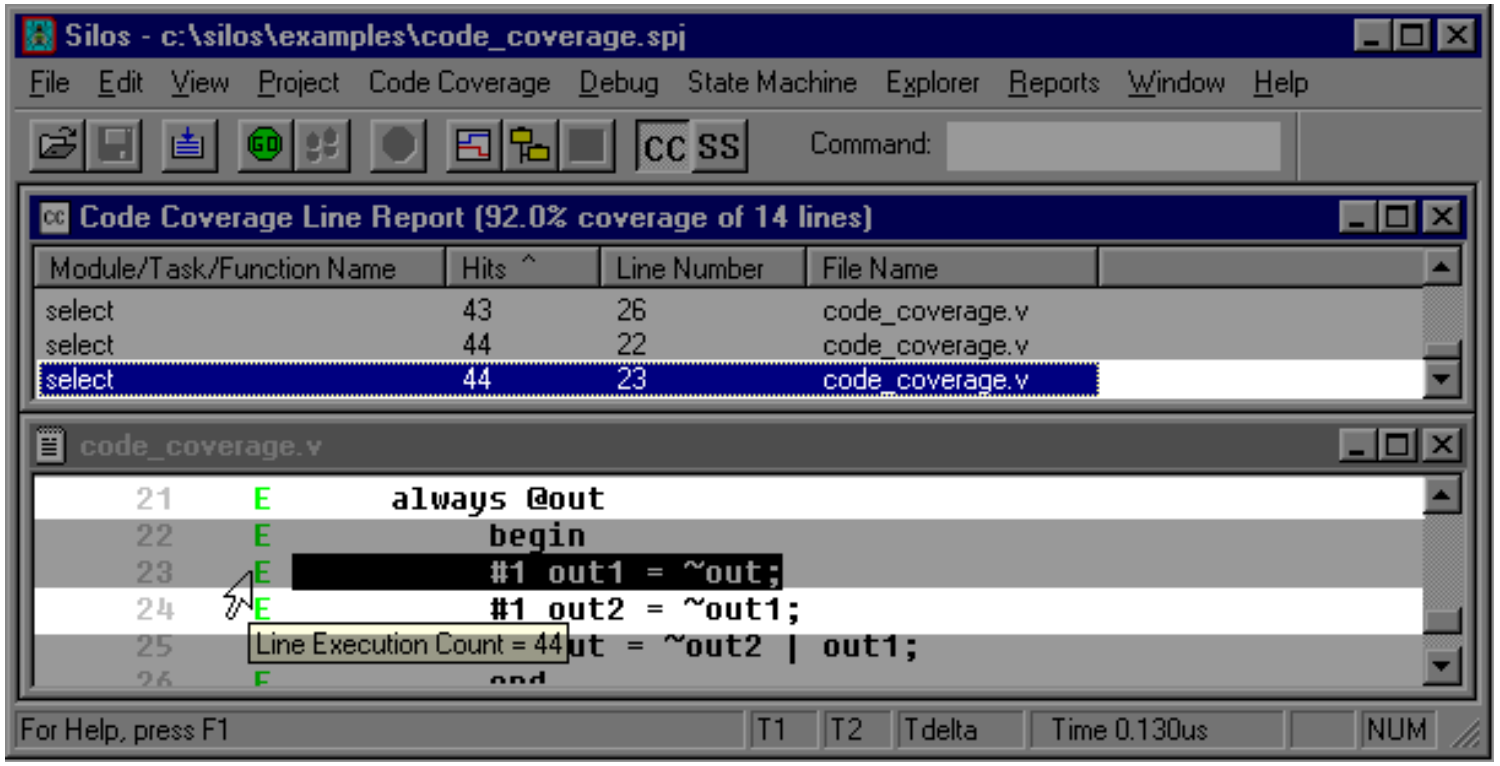
Code Coverage



Code Coverage button should be depressed.
Double click on the first line in the Operator report [1] to open the source file at that line. Lines not executed have a purple dot [2].
Click on a column header [3] to change the sort order.

SILVACO

Code Coverage

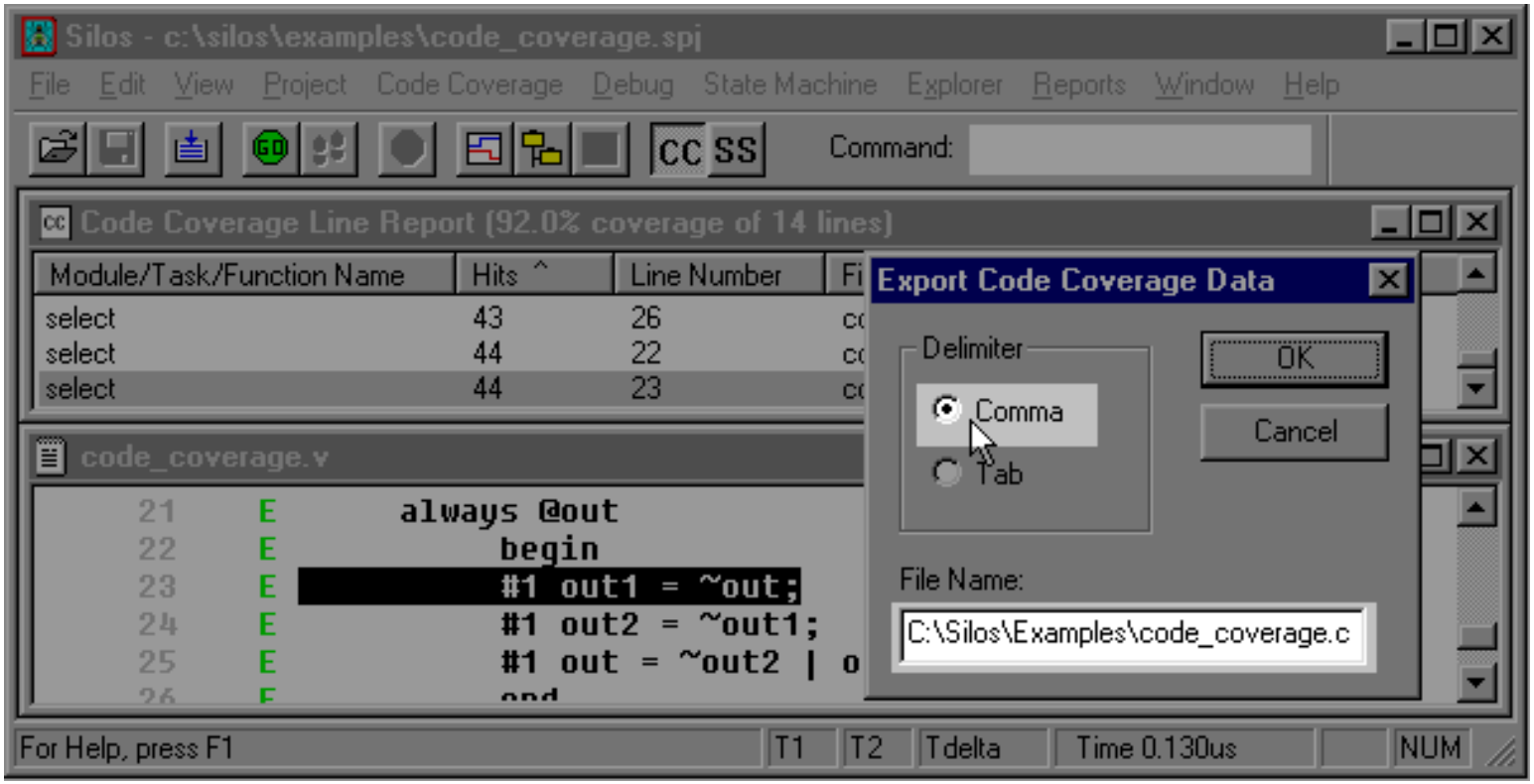


A large number of “hits” in the “Code Coverage Line Report” may indicate a looping problem in the user’s design, such as:

```
always @out
out = ~out2 | out1;
```

SILVACO

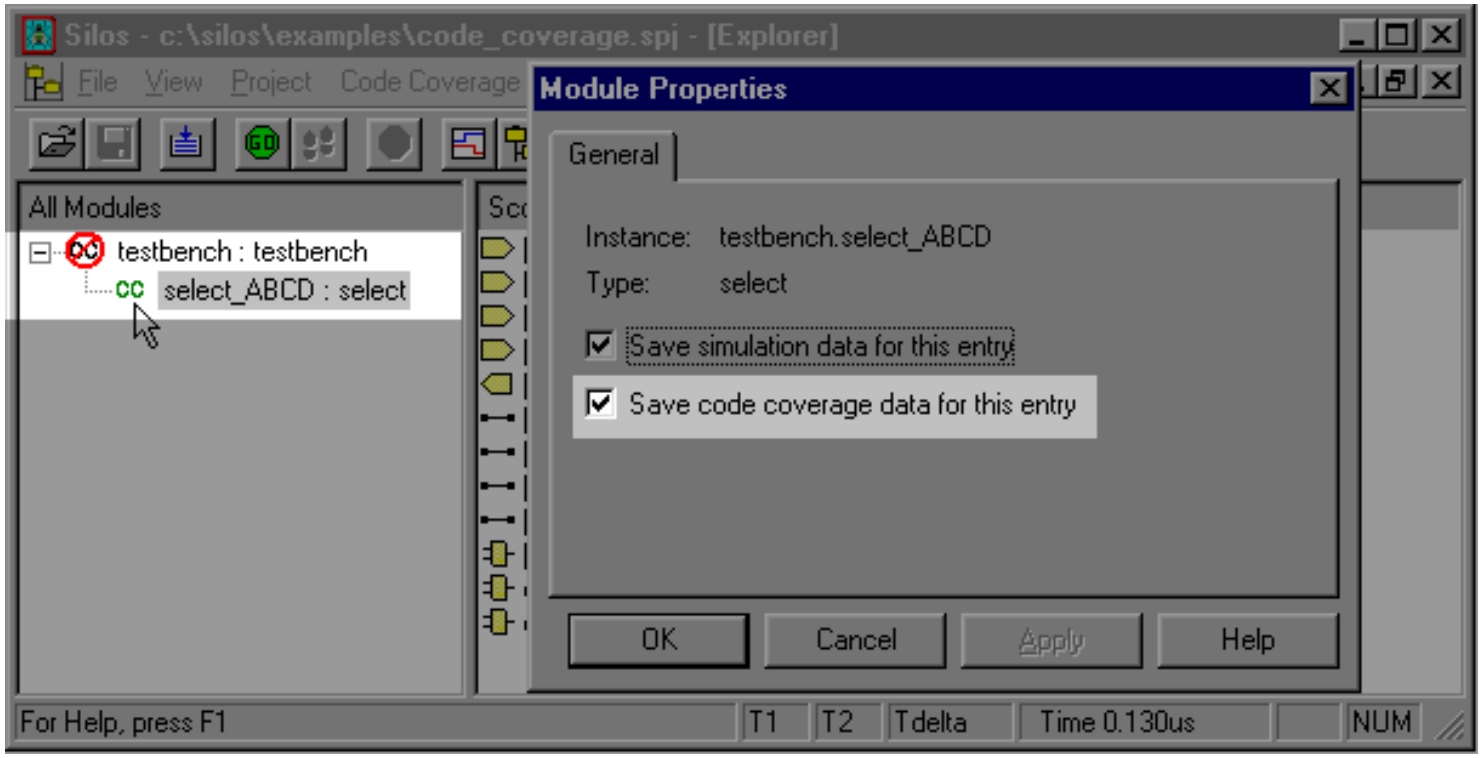
Code Coverage



"Export Code Coverage Data" dialog box can use comma separated data files for importing into a spreadsheet program, such as Microsoft Excel.

SILVACO

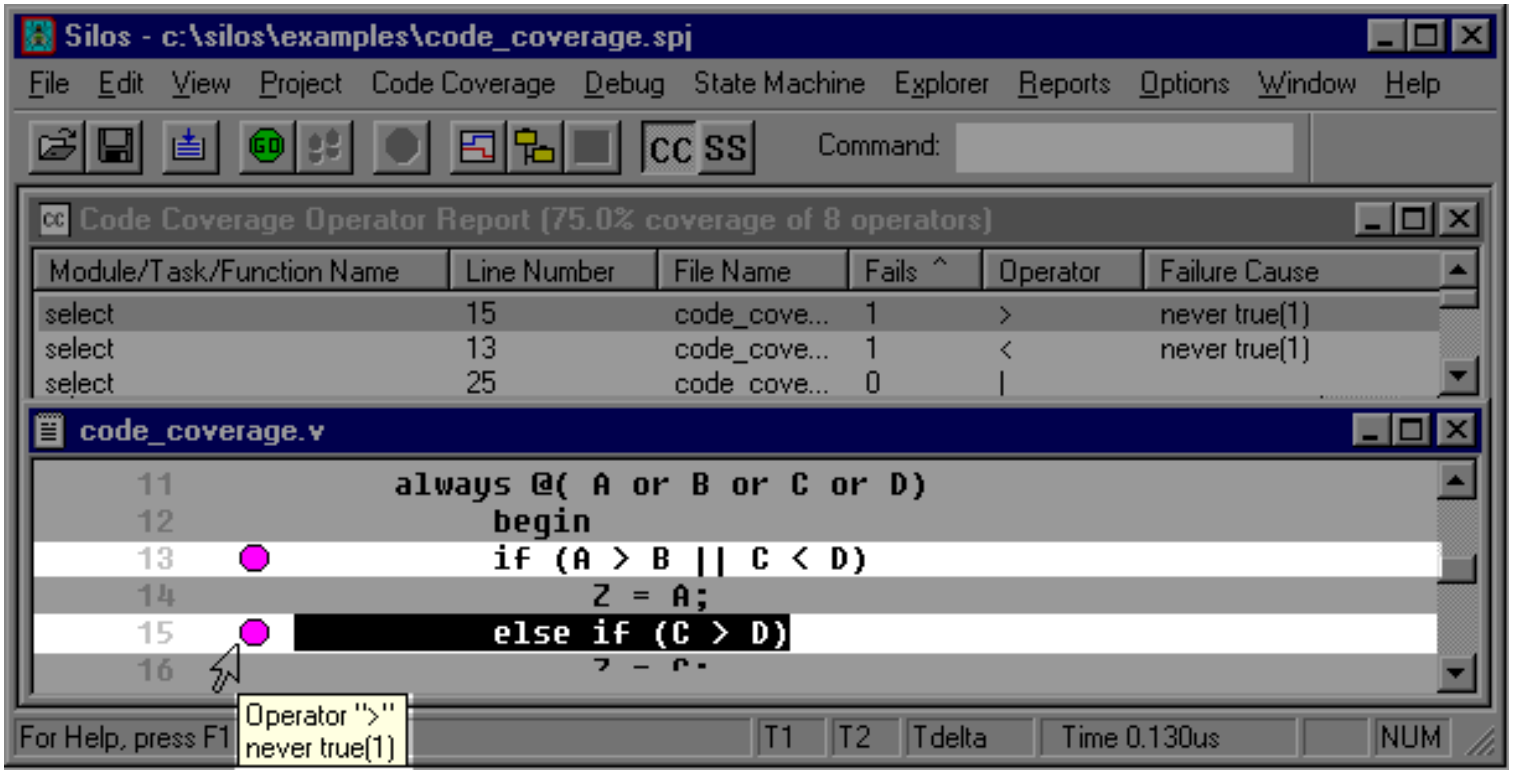
Operator Code Coverage



In the Explorer, the red circle with the slash through it means code coverage is disabled for this instance. The green “CC” means code coverage is enabled.

SILVACO

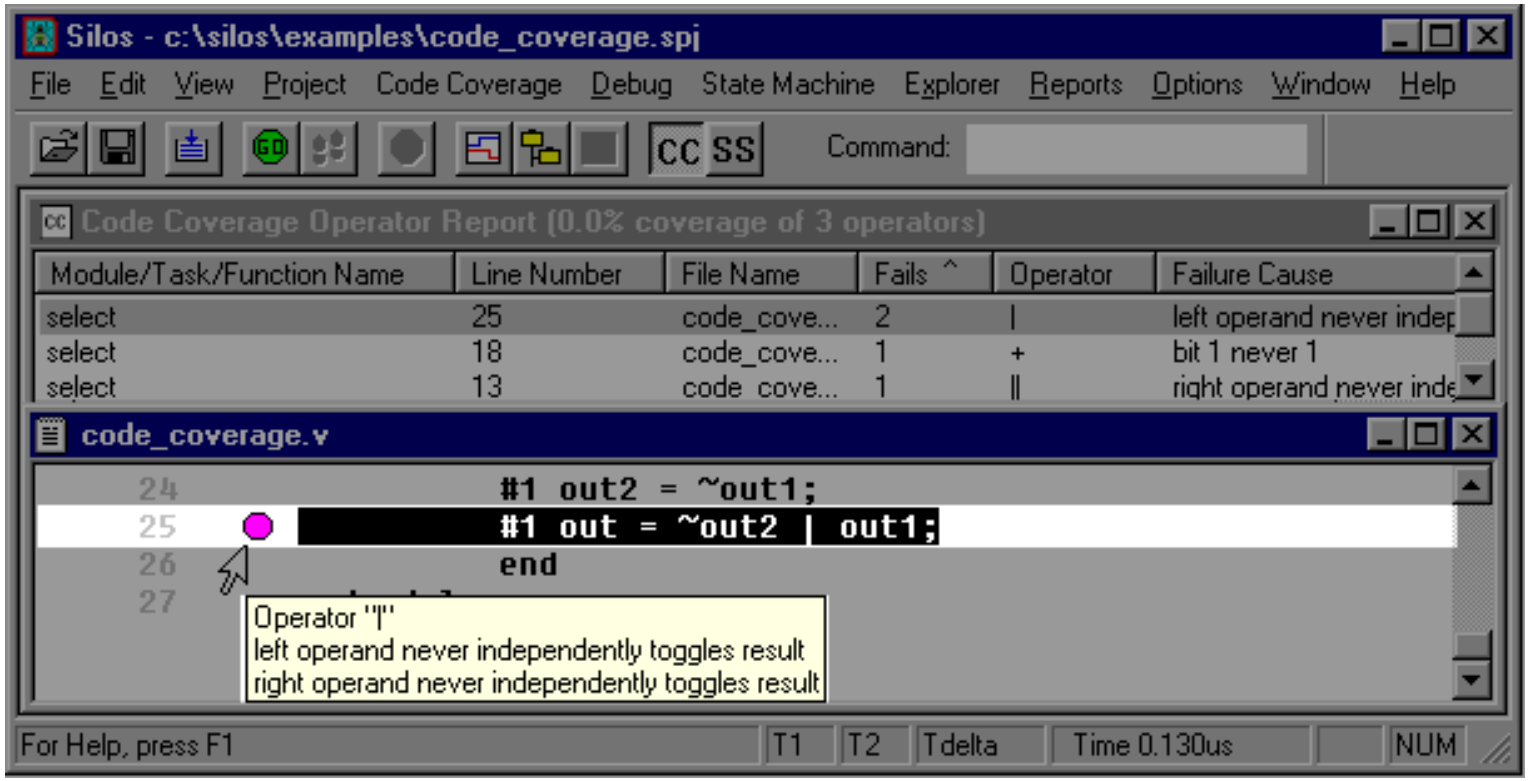
Operator Code Coverage



The “<” and the “>” operators failed to be true. When expressions have more than one operator, you can use separate lines to display the purple dot for each failed operator.

SILVACO

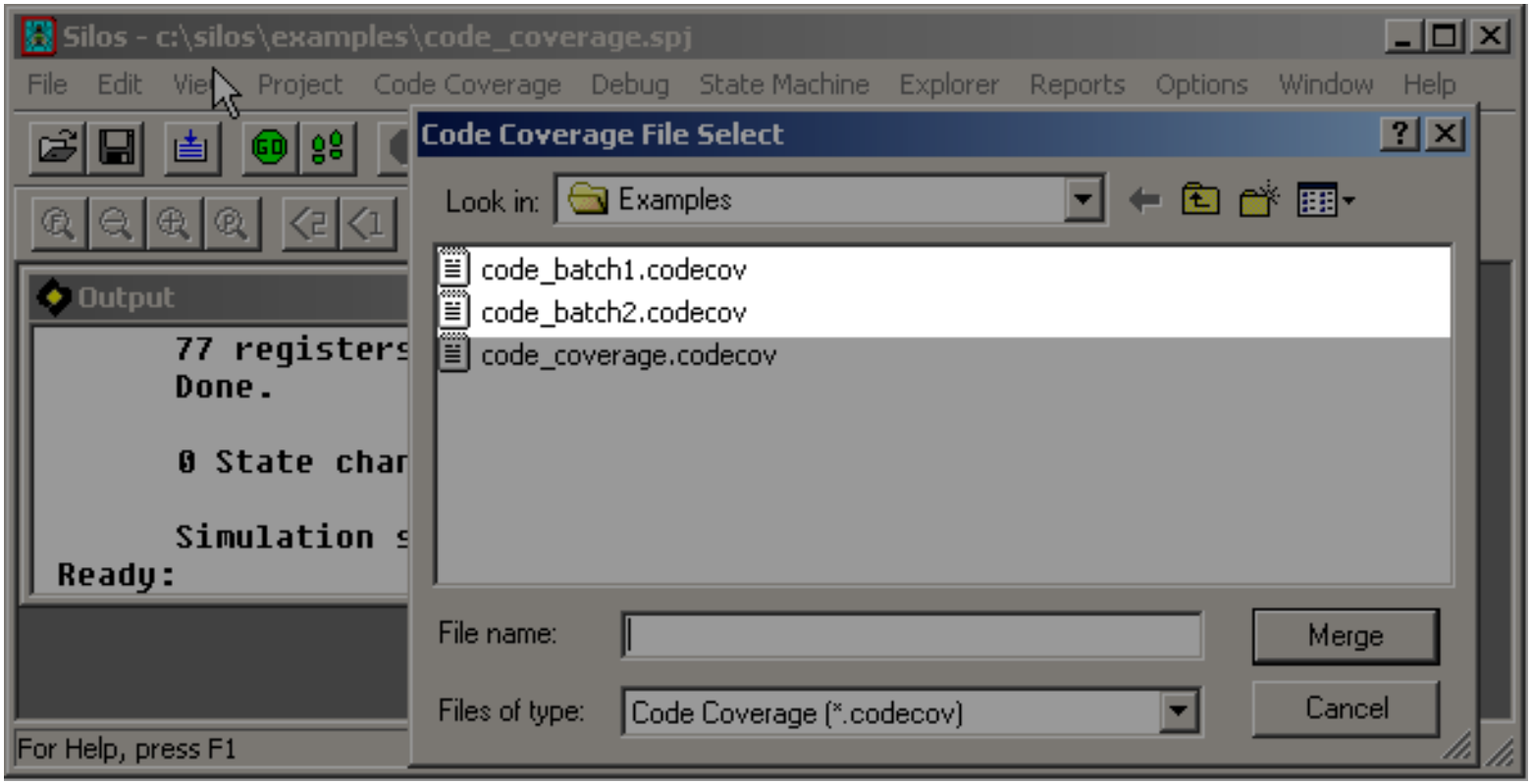
Operator Code Coverage


Silos - c:\silos\examples\code_coverage.spj
File Edit View Project Code Coverage Debug State Machine Explorer Reports Options Window Help
Command:
Code Coverage Operator Report (0.0% coverage of 3 operators)
Module/Task/Function Name | Line Number | File Name | Fails | Operator | Failure Cause
select | 25 | code_cove... | 2 | | | left operand never indep
select | 18 | code_cove... | 1 | + | bit 1 never 1
select | 13 | code_cove... | 1 | || | right operand never inde
code_coverage.v
24 #1 out2 = ~out1;
25 #1 out = ~out2 | out1;
26 end
27
Operator '|'
left operand never independently toggles result
right operand never independently toggles result
For Help, press F1
T1
T2
Tdelta
Time 0.130us
NUM

Deselecting the "DEFAULT" option, and selecting the other options for the Operator report will show which operands did not affect their corresponding operator.

SILVACO

Merging Code Coverage



Code coverage results can be merged from batch simulations that use different test benches for the same design.

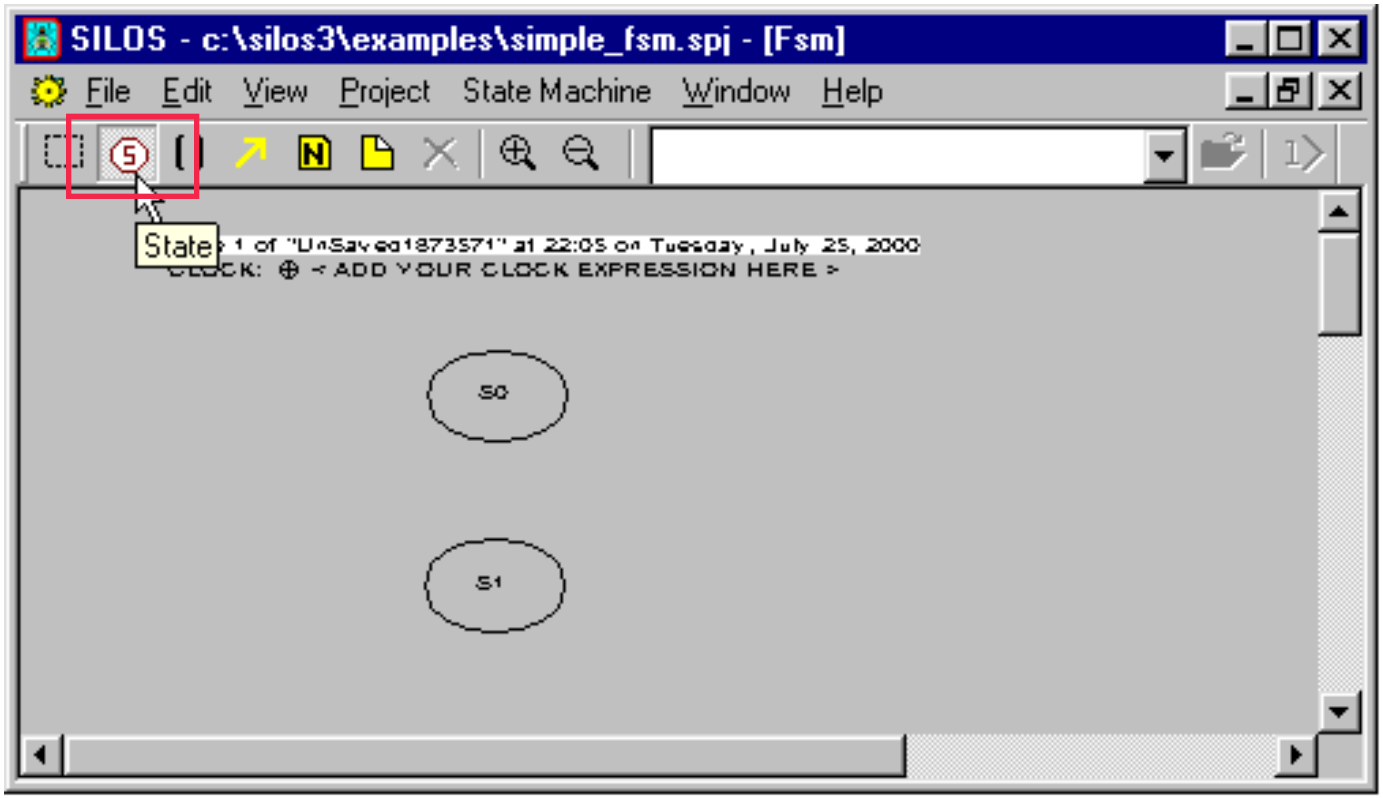
SILVACO

Agenda: Finite State Machine Example

- Introduction
- Starting Silos Project
- Explorer and Analyzer
- Source Code Debugging
- State Machine Design Entry
- Advanced Debugging Features
- Finite State Machine Example
- Gate Level Debugging

SILVACO

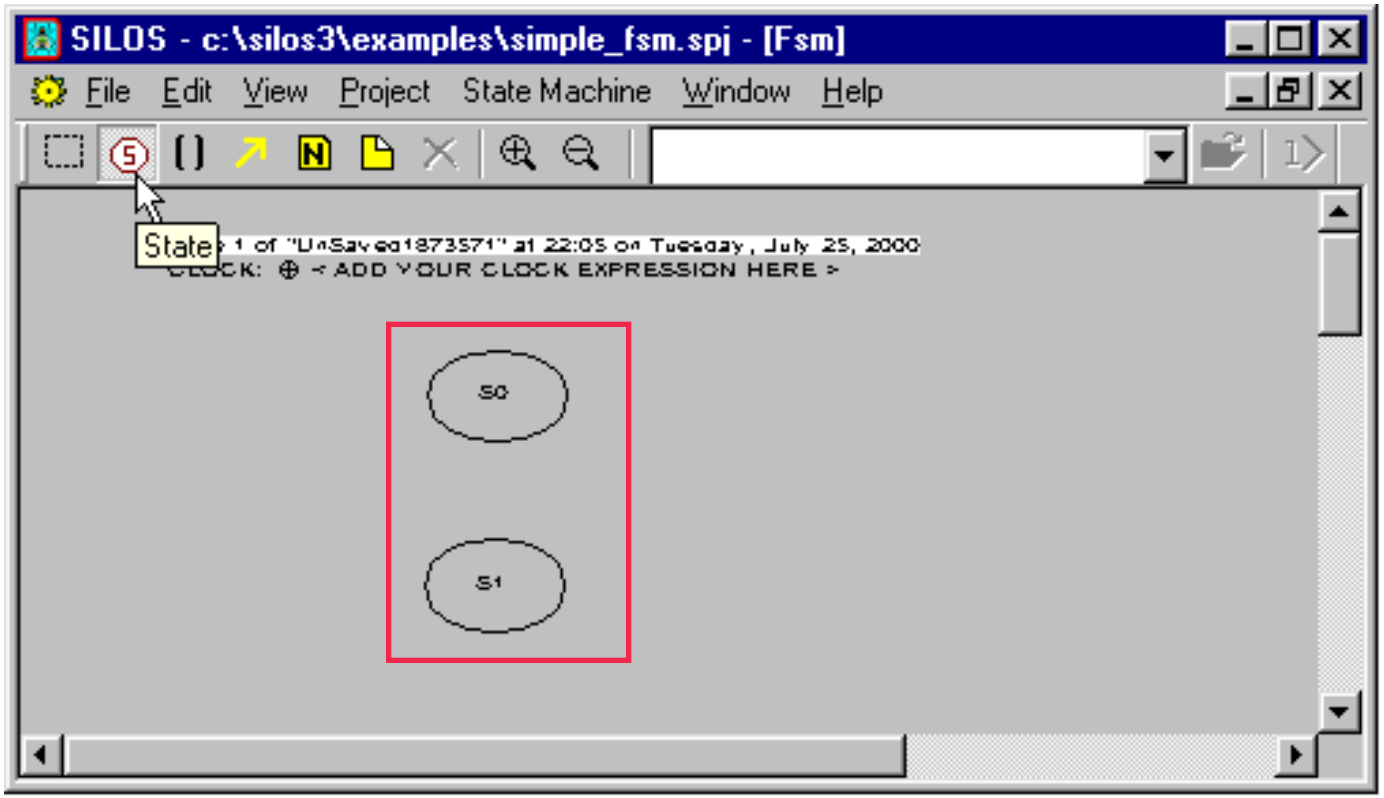
Finite State Machine (FSM) Entry



FSM Toolbar. State drawing button.

SILVACO

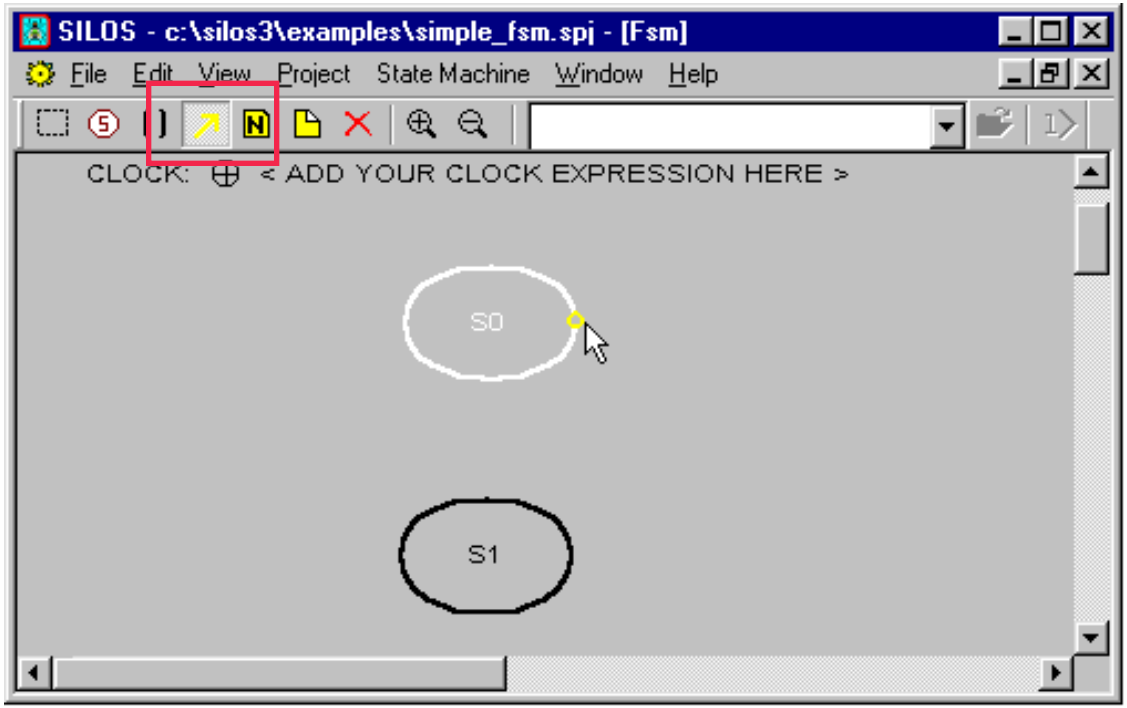
Finite State Machine (FSM) Entry



Use the left mouse button when the State Mode is active to create the two states.

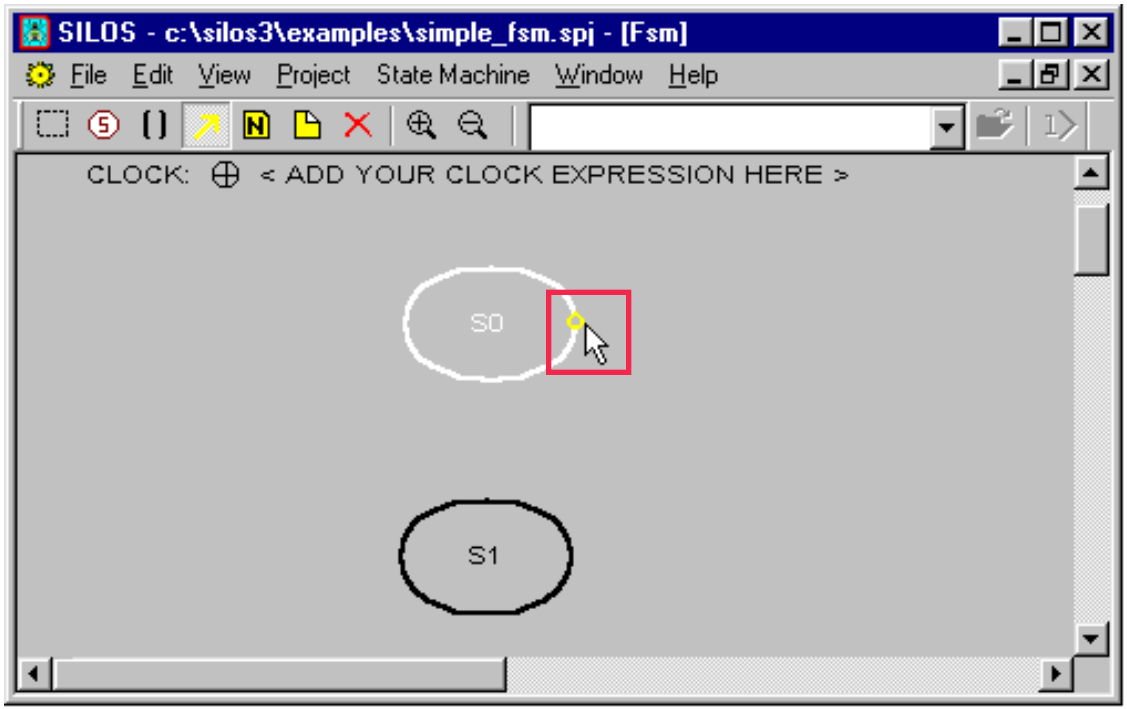
SILVACO

FSM Entry - Transition Mode



Transition Mode button.

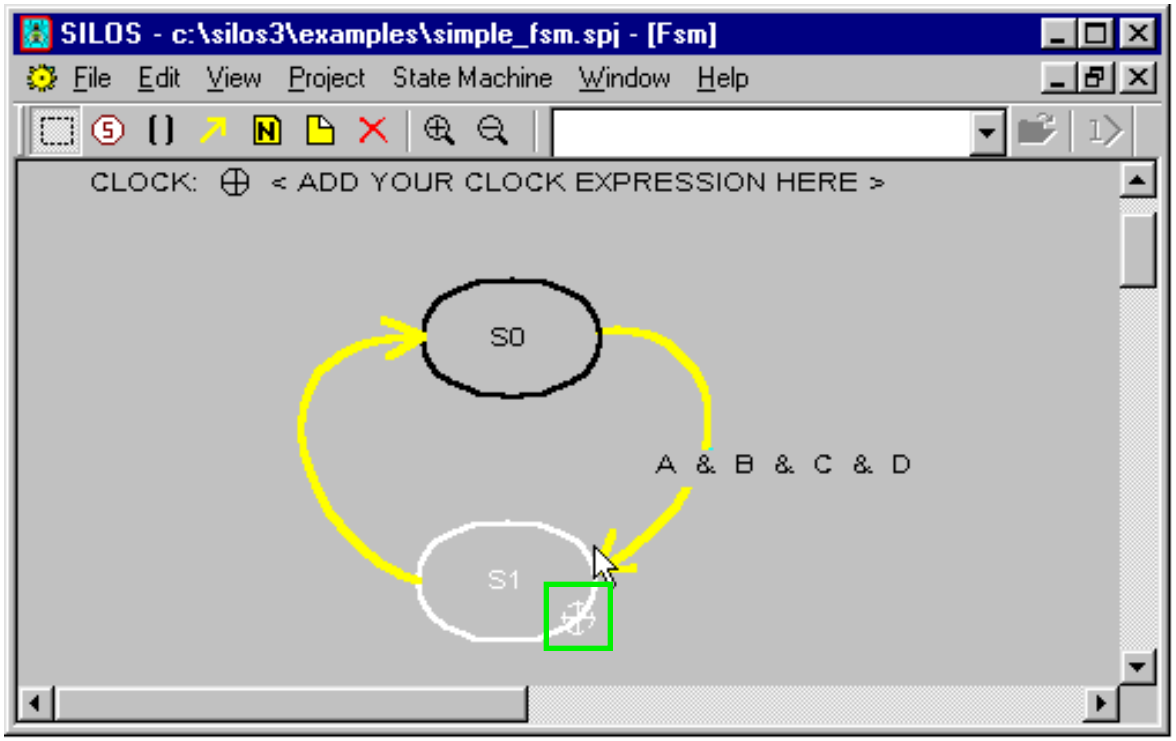
FSM Entry - Transition Mode



Notice the small circle on state “s1” for drawing the transition between “s1” and “s2”. The circle will move along “s1” as the mouse is moved.

SILVACO

FSM Entry - Transition Mode

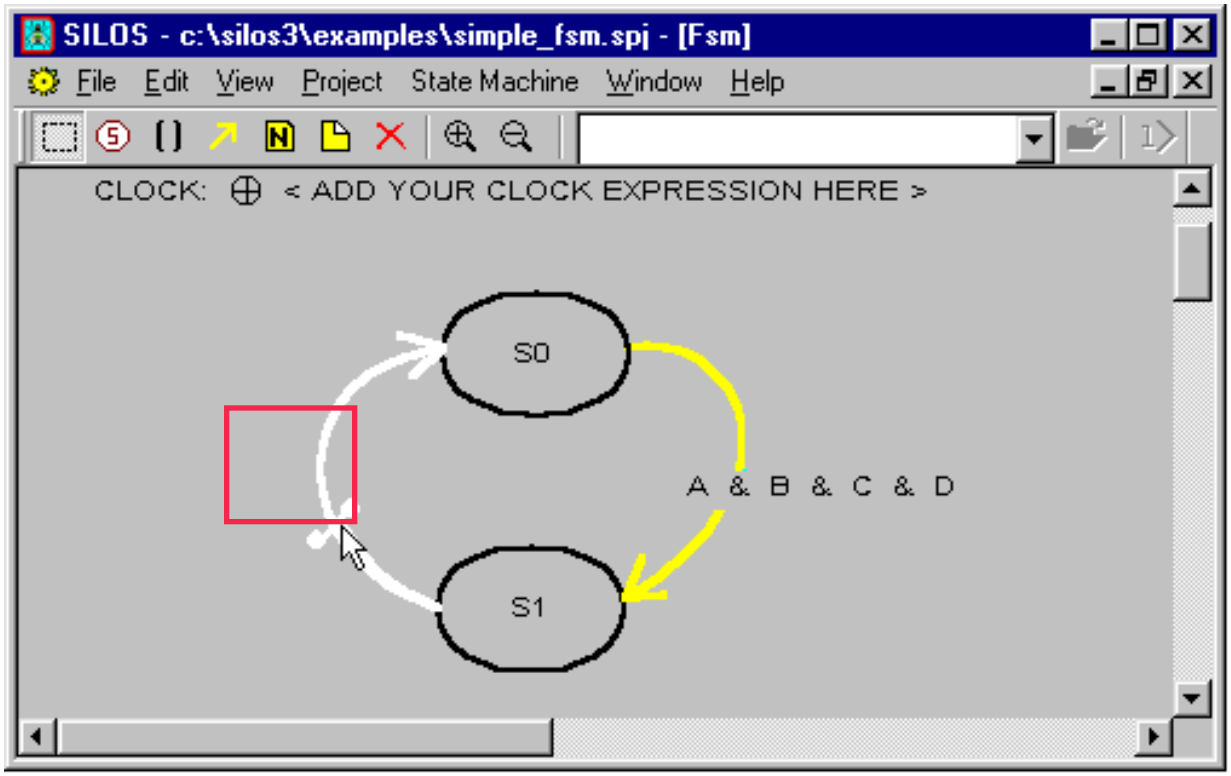


Transition between states “s2”and “s1” will change at the next positive clock edge because the expression has been deleted.

Notice a state symbol can be moved when there is a circle with cross hairs. If instead you see a star on the state symbol, this will change the shape of the state symbol instead of moving it.

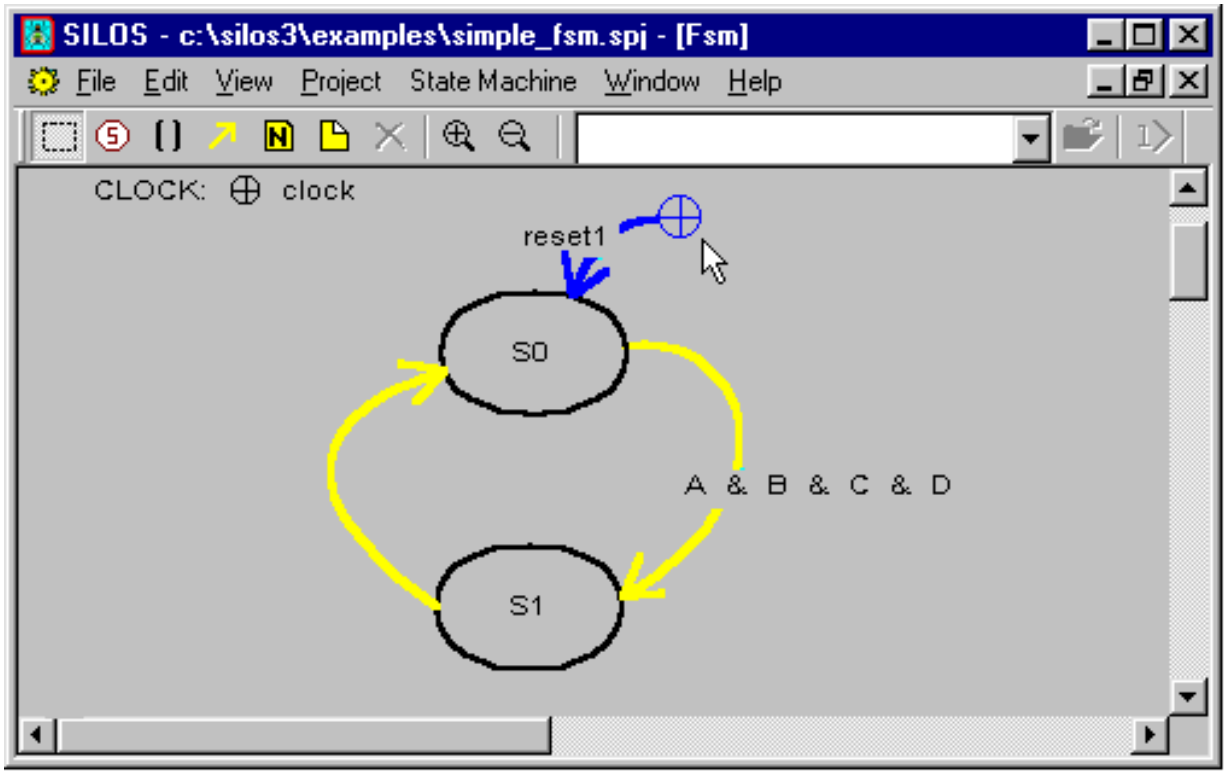
SILVACO

FSM Entry - Transition Mode



The transition can be redrawn by moving the mouse cursor tangentially across the transition line with the left mouse button held down.

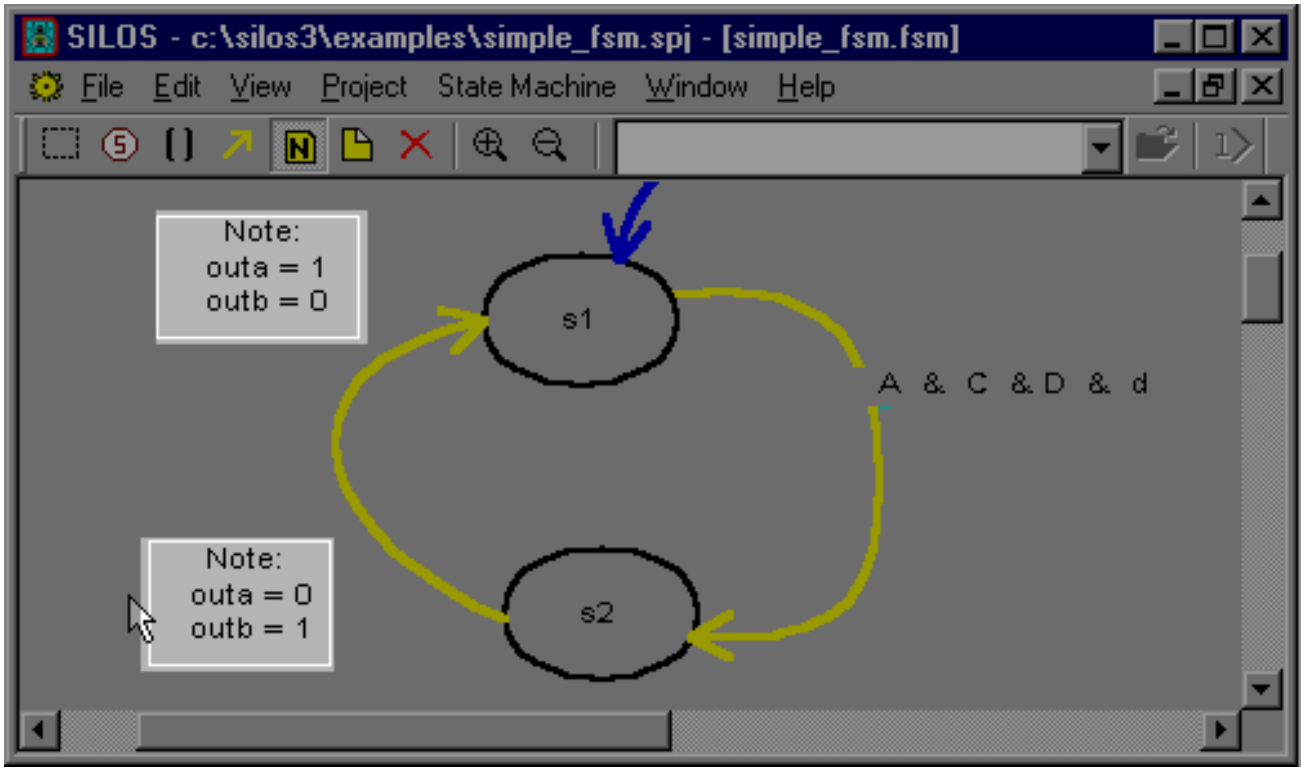
FSM Entry - Transition Mode



The "+" symbols are for positive edge clock and reset lines. Click on the "+" sign to change it to a "-".sign for a negative edge reset clock.

SILVACO

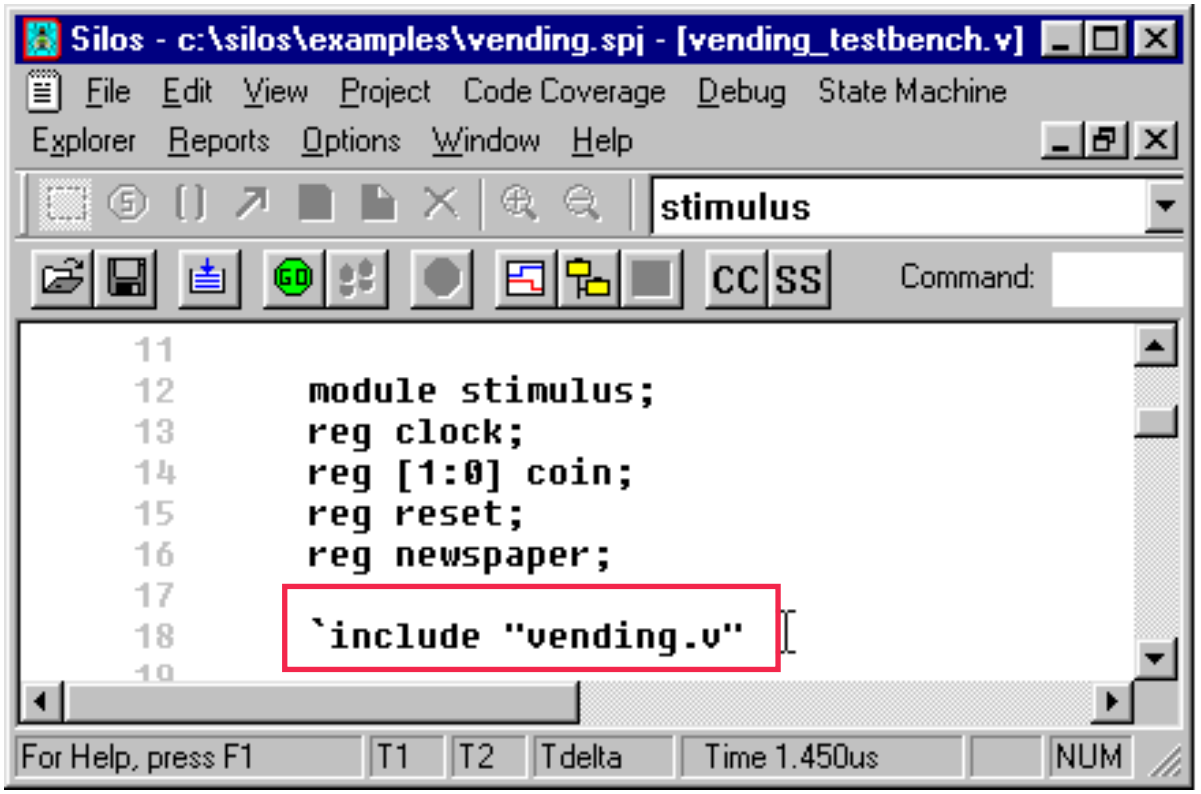
FSM Entry - Notes



Notes can be used to document output signals on the diagram.

SILVACO

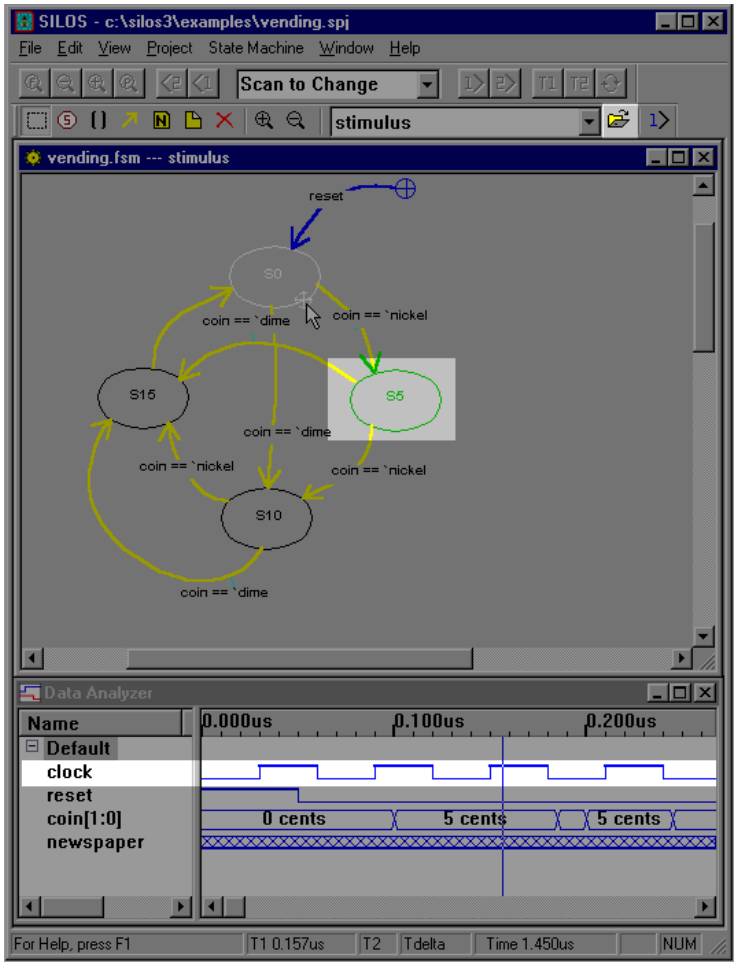
FSM Debugging



The `include “vending.v” compiler directive is used to include the Verilog HDL source code for the newspaper vending FSM.

SILVACO

FSM Debugging

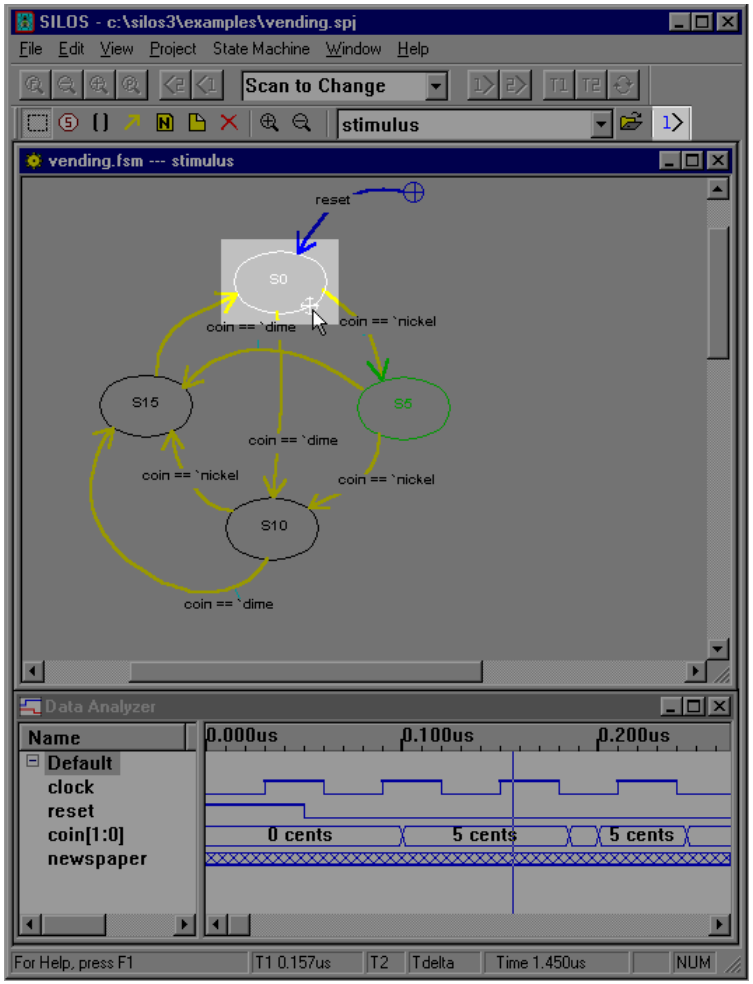


Open Instance button

As the “T1” timing marker is dragged in the Data Analyzer, the states change color as they become active.

SILVACO

FSM Debugging

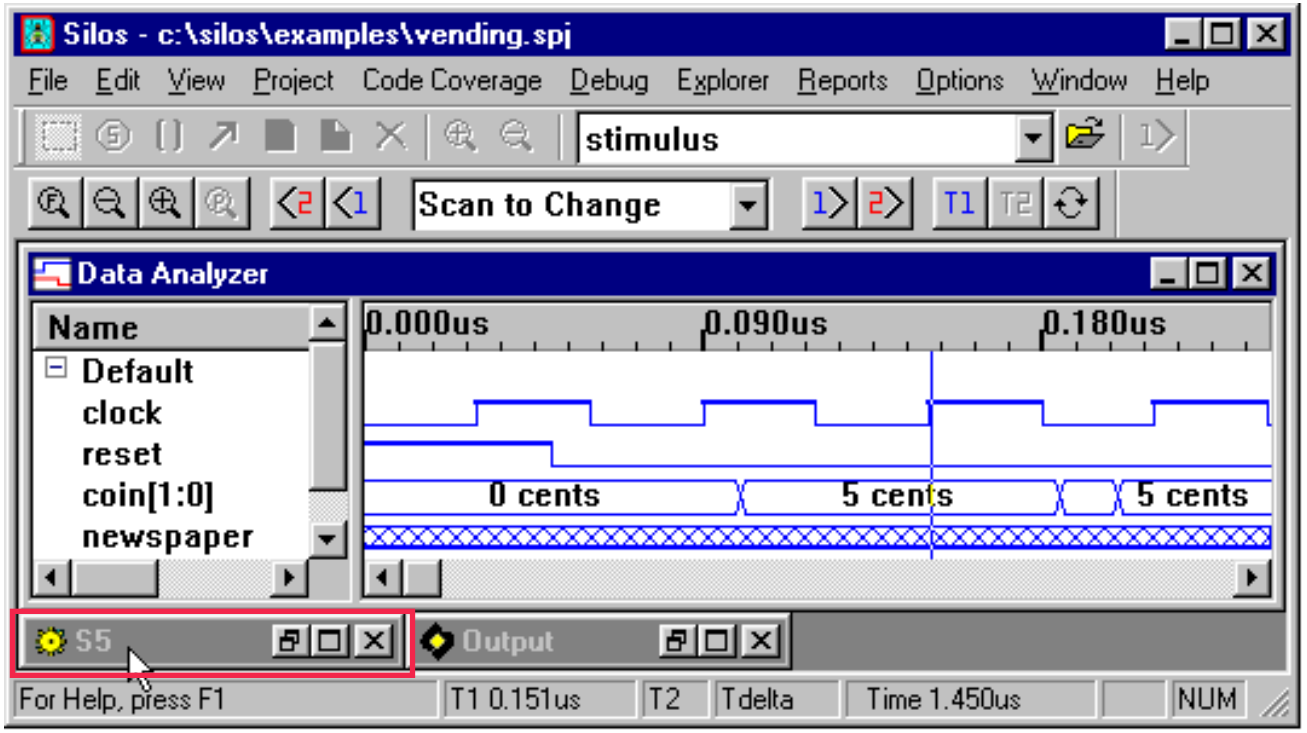


"FSM Scan" button

To select states for scanning, click on the state with the left mouse button and hold down the "Ctrl" key.

SILVACO

FSM Debugging - Finite State Machine Window



The minimized Finite State Machine window shows the active state as you scan in the Data Analyzer.

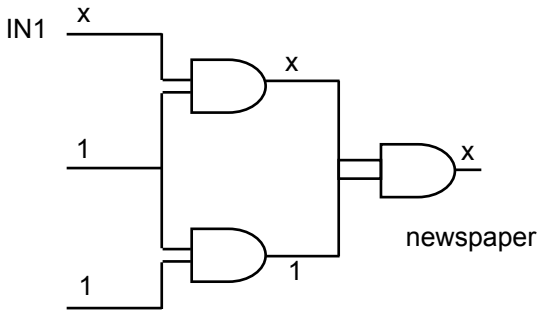
SILVACO

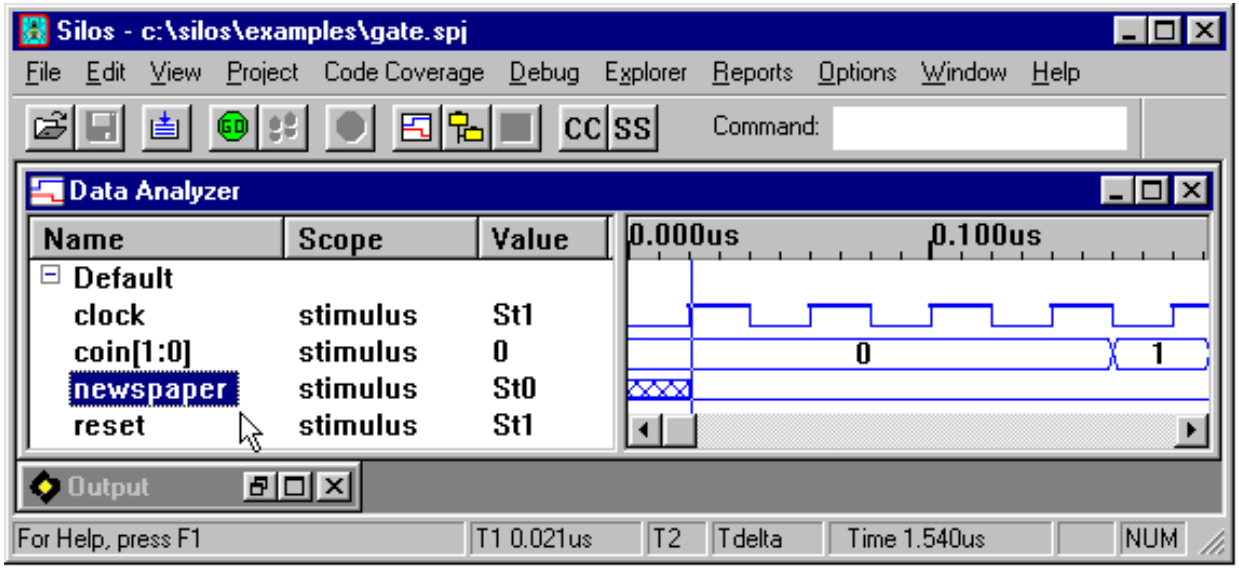
Agenda: Gate Level Debugging

- Introduction
- Starting Silos Project
- Explorer and Analyzer
- Source Code Debugging
- State Machine Design Entry
- Advanced Debugging Features
- Finite State Machine Example
- Gate Level Debugging

SILVACO

Gate Level Debugging



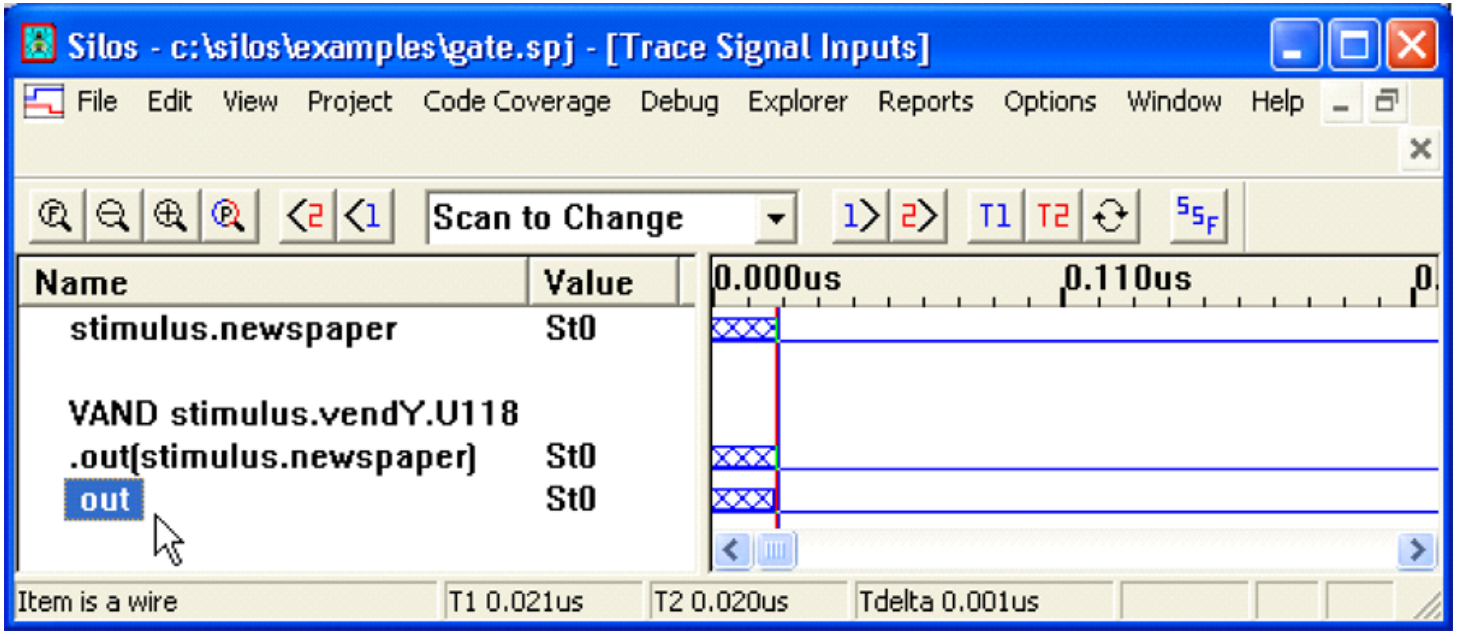


Highlight signal "newspaper" and click with the right mouse button in the Signal Name list box. Select "Trace Signal Inputs" when the context menu opens.

At time=0.021us, "newspaper" changes from Unknown to Low.

SILVACO

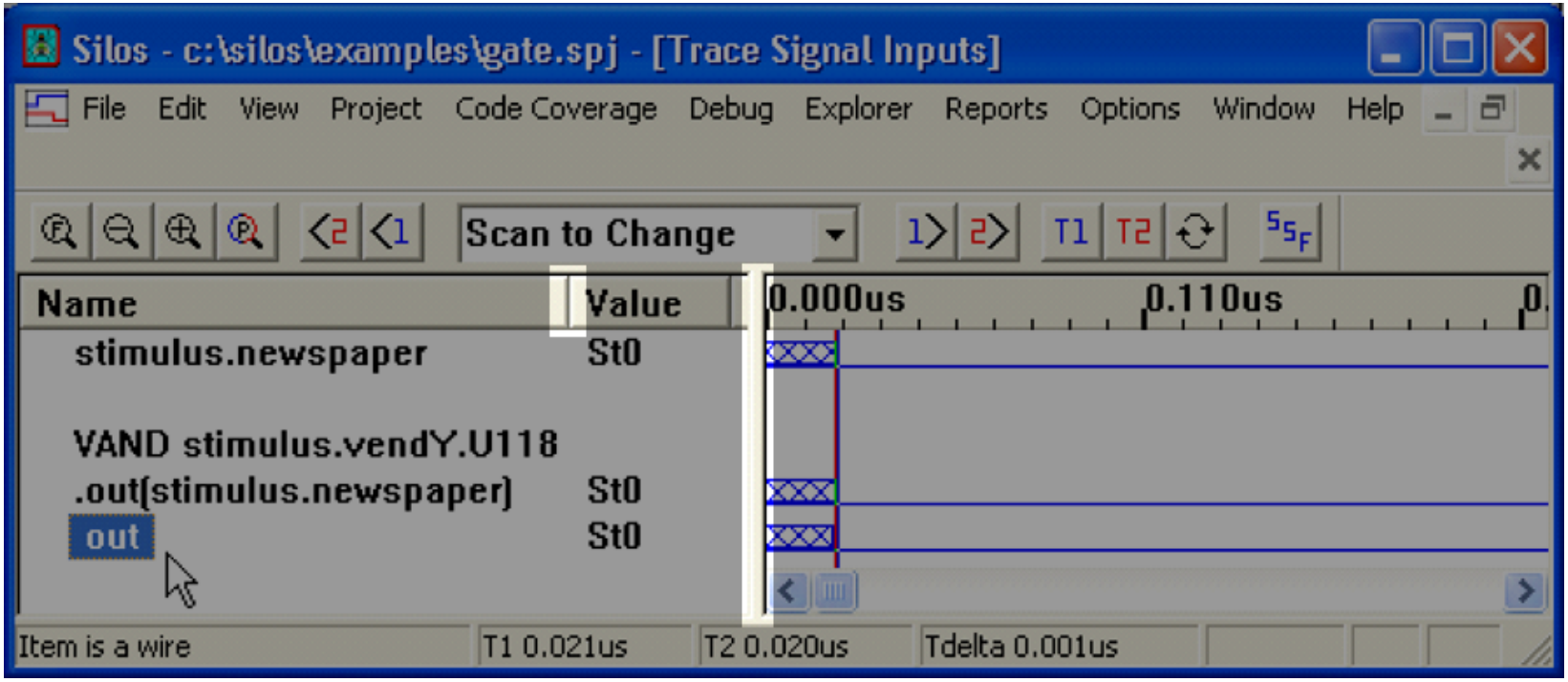
Gate Level Debugging



The top signal, “stimulus.newspaper”, is the signal being traced. The next line is the module name, “VAND”, and instance name, “stimulus.vendY.U118”, which drives the top signal. The next line is the driver in the instance. Subsequent lines indented with a space are the inputs to the driver. If the driver is a port of the instance it is shown using port name syntax. If it is not a port then just the name of the net is listed. The signal name preceded by the period, “.out”, is the local port name in module “VAND”. The name in the parenthesis, “(stimulus.newspaper)”, is the wire name in the module above “stimulus.vendY.U118”. Signal name “out” is listed below it and indented one space.

SILVACO

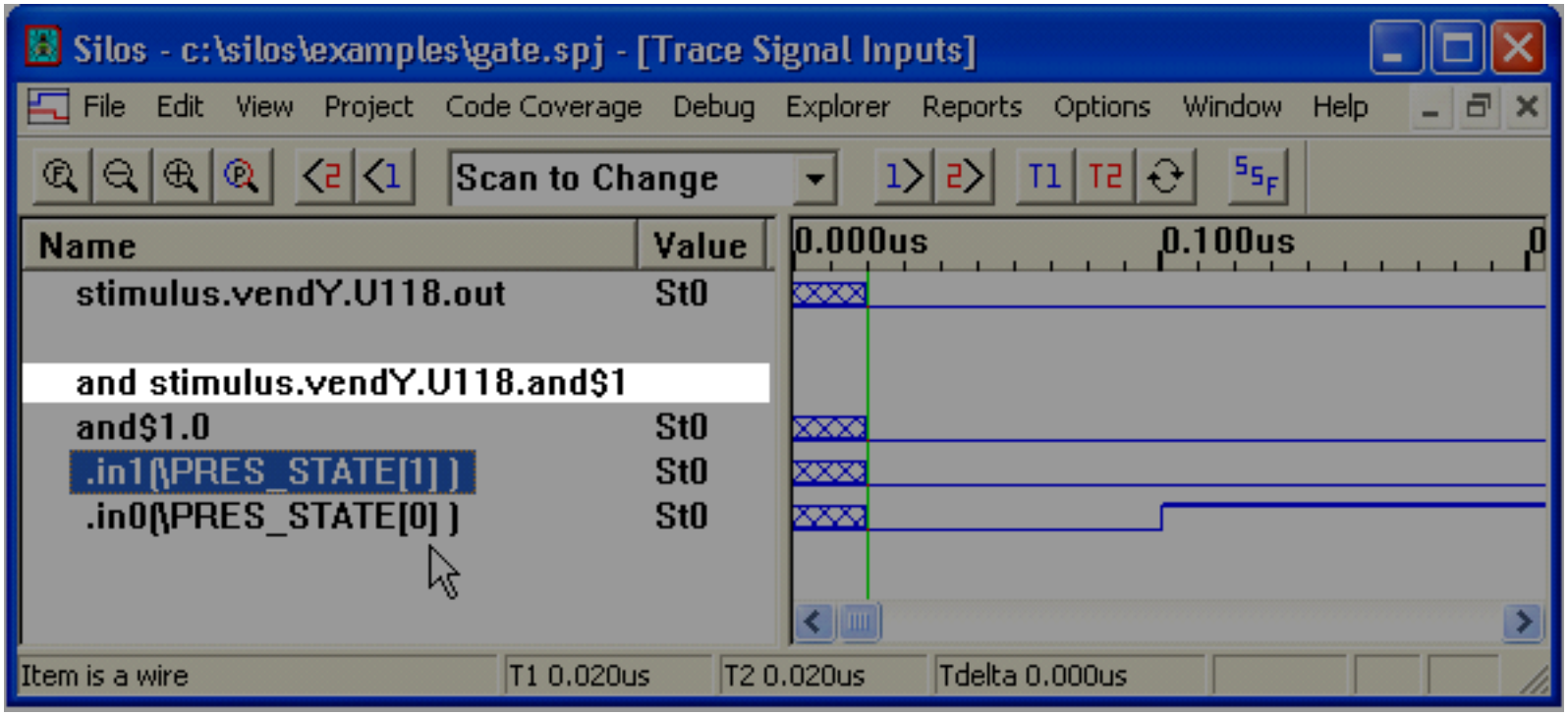
Gate Level Debugging- Resize Interface



To view the full signal name, use the mouse to drag the vertical bars and increase the size of the Name column.

SILVACO

Gate Level Debugging

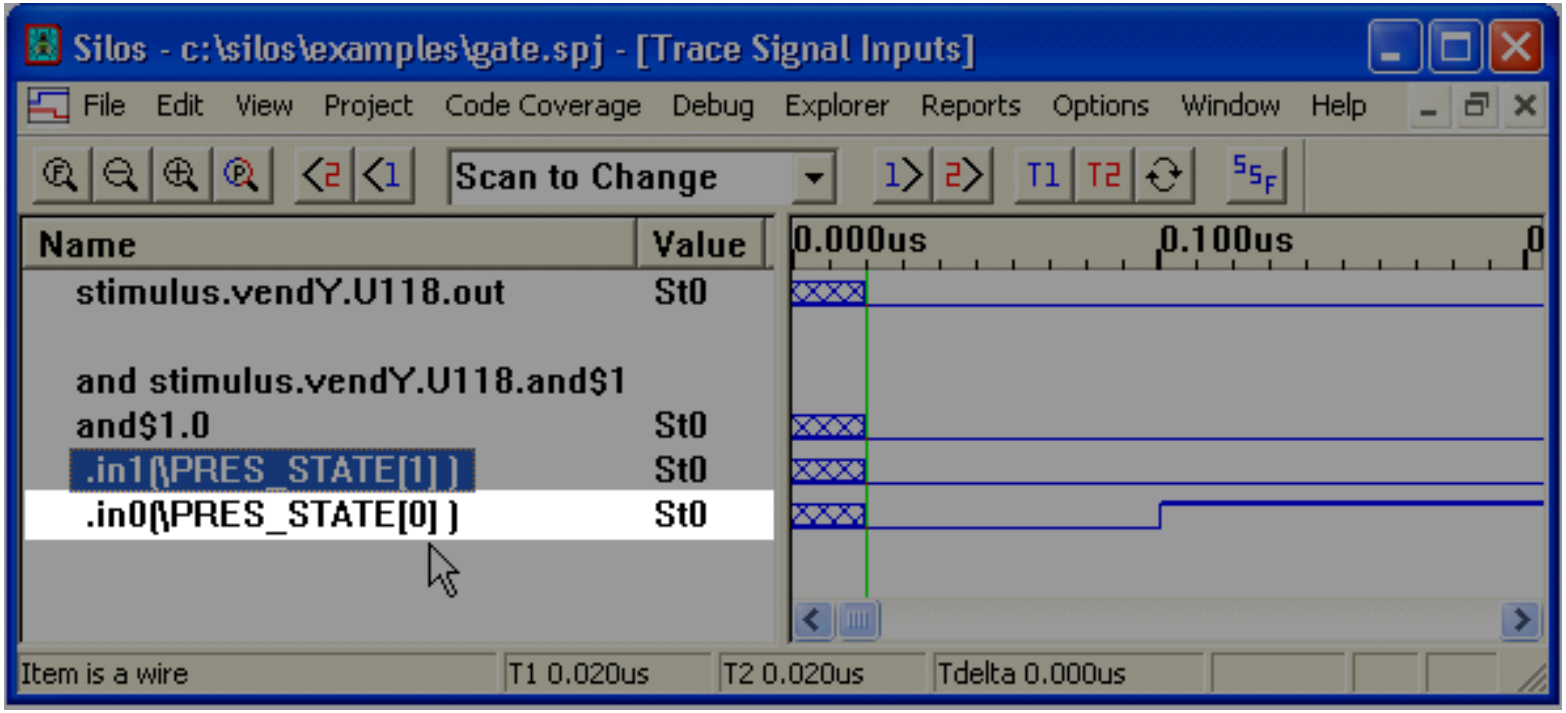


Two input "and" gate.

Gate instance name. The $1 at the end of the name is an IEEE naming convention for gates that do not have instance names.

SILVACO

Gate Level Debugging

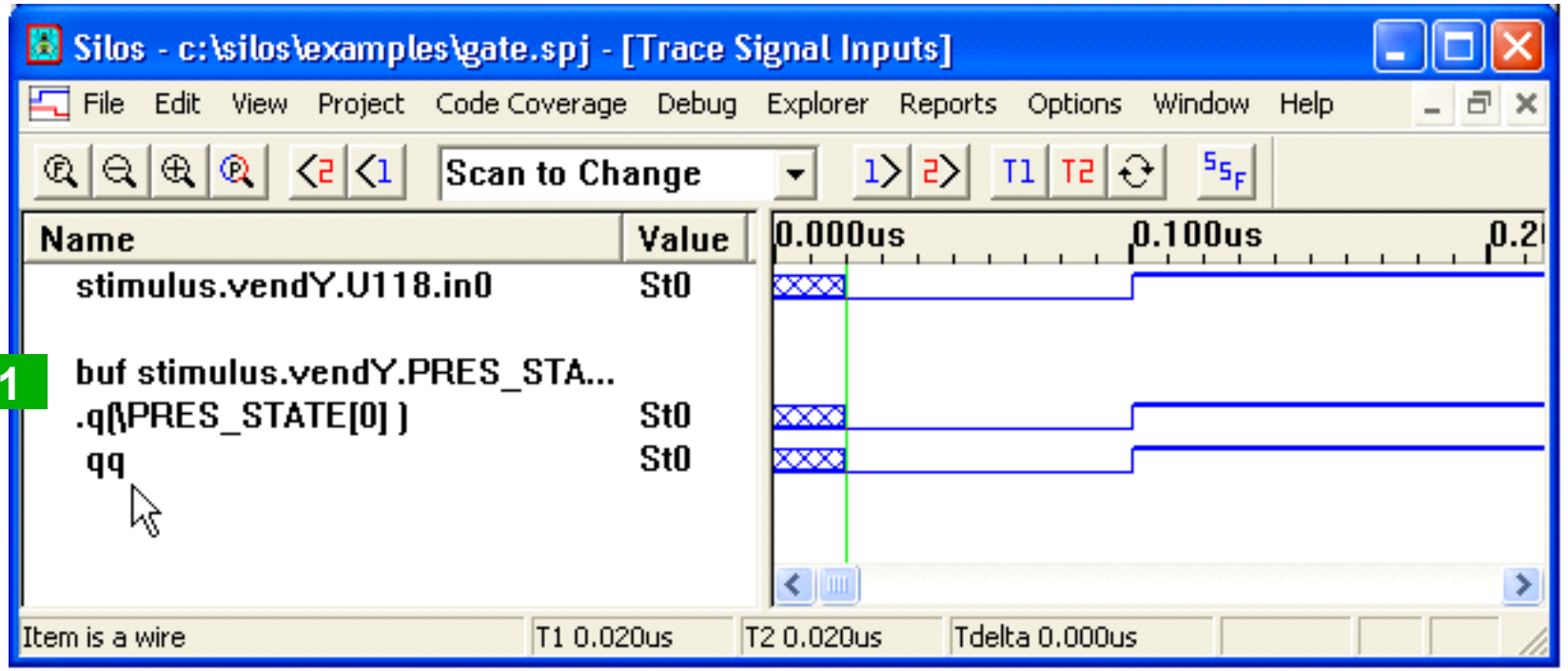


"in0" is the local name for the input pin. "\ PRES_STATE[0]" is the wire name in the module above "stimulus.vendY.U118".

Since both inputs change from Unknown to Low at time=0.020us for this example, double click on input "in0".

SILVACO

Gate Level Debugging - Buffer Gate

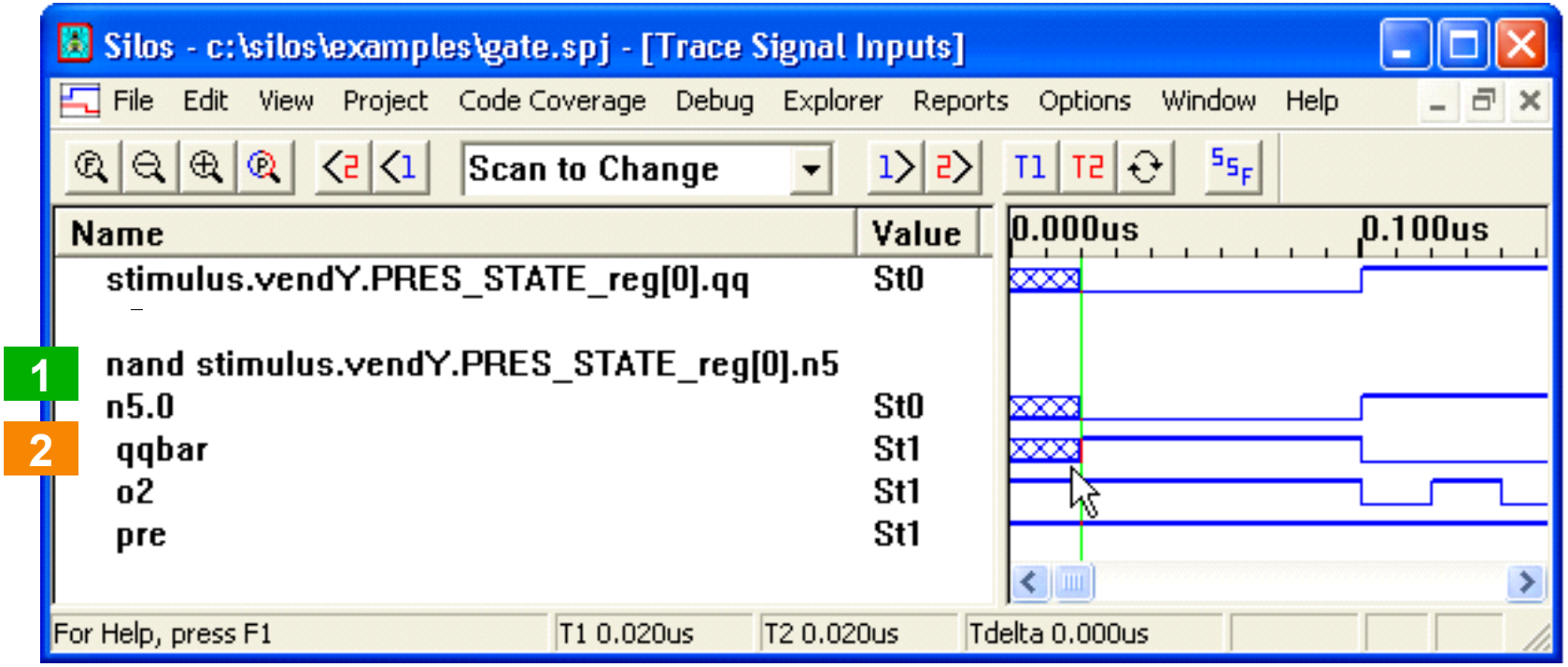


Buffer gate [1]

Double-click on the input to the buffer to continue tracing back through the topology.

SILVACO

Gate Level Debugging - Buffer Gate - “qqbar”

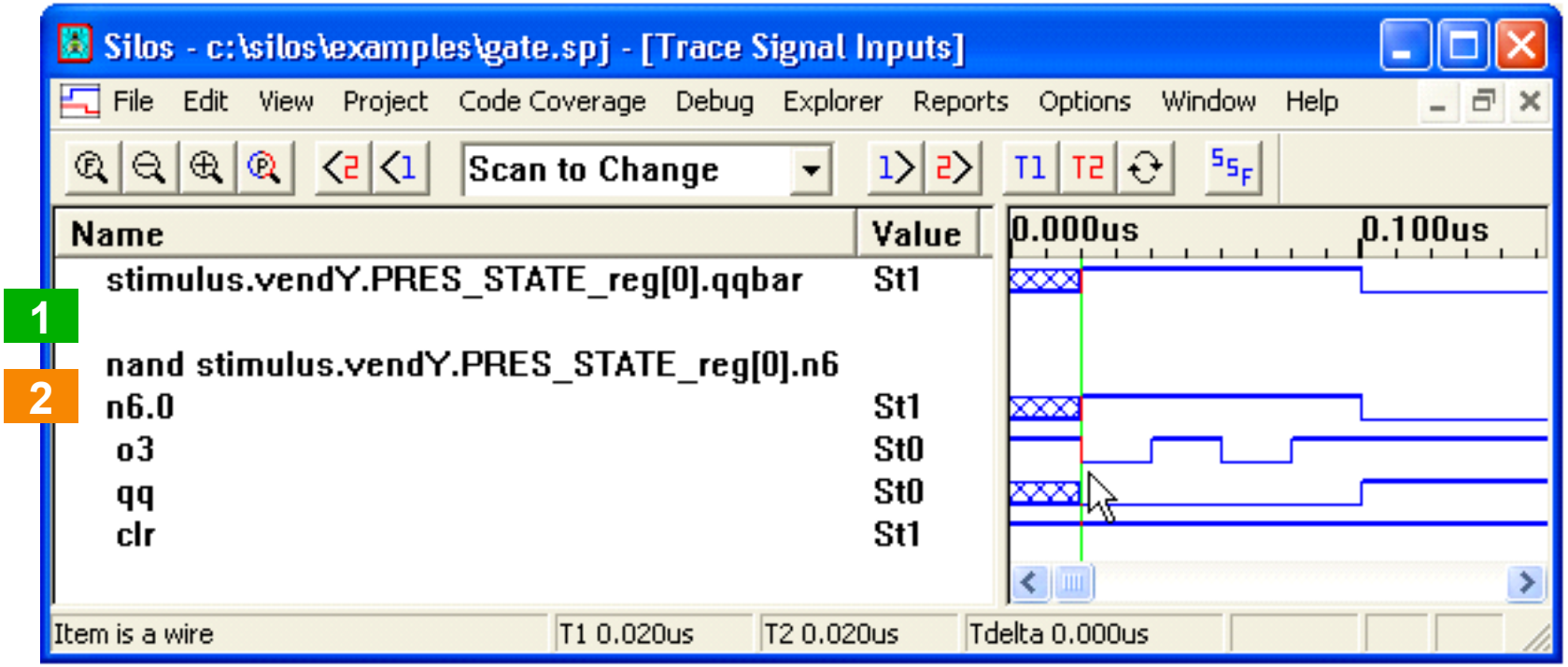


Three input “nand” gate [1].

Double-click on the “qqbar” [2] input because the Unknown to High transition on this input caused the output to change.

SILVACO

Gate Level Debugging - Buffer Gate - “o3”

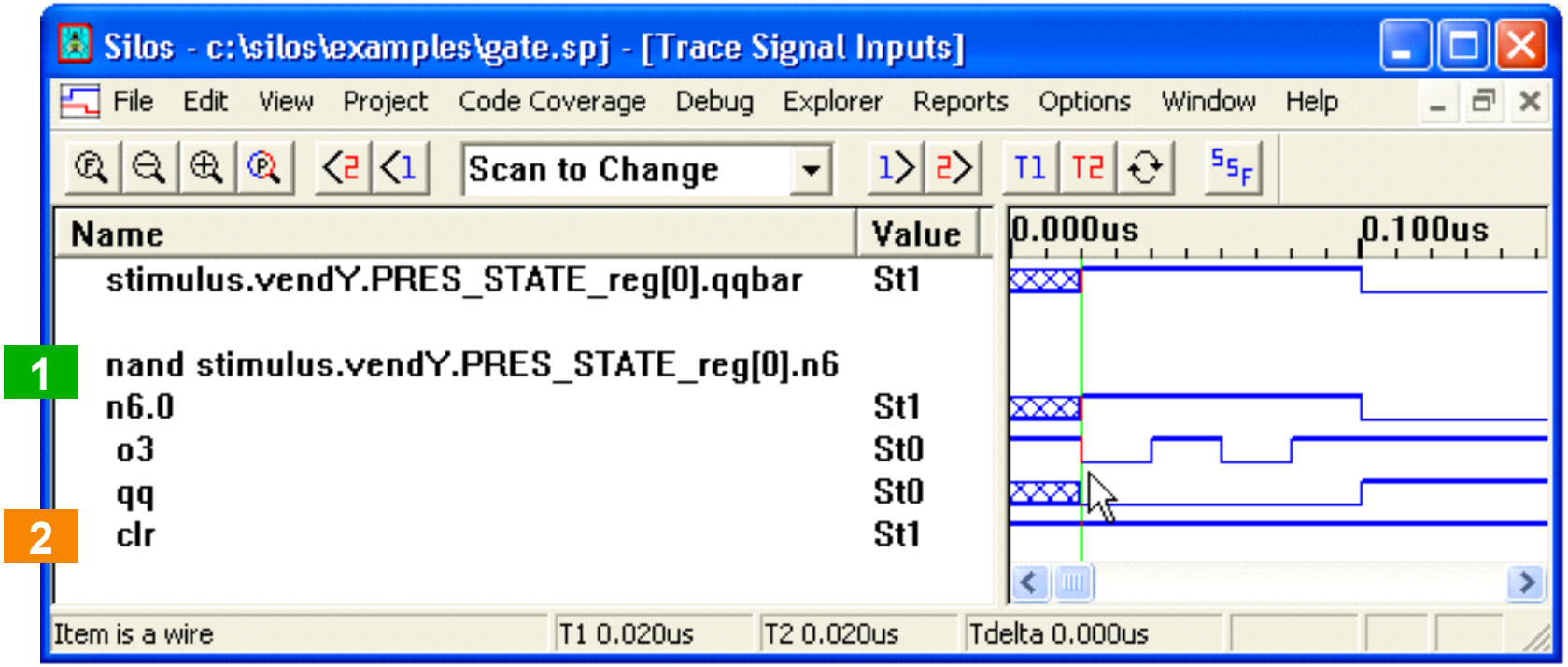


Three input “nand” gate [1].

Double-click on the “o3” [2] input to the “nand” gate because the High to Low transition on this input caused the output to change.

SILVACO

Gate Level Debugging - Buffer Gate - “ck”

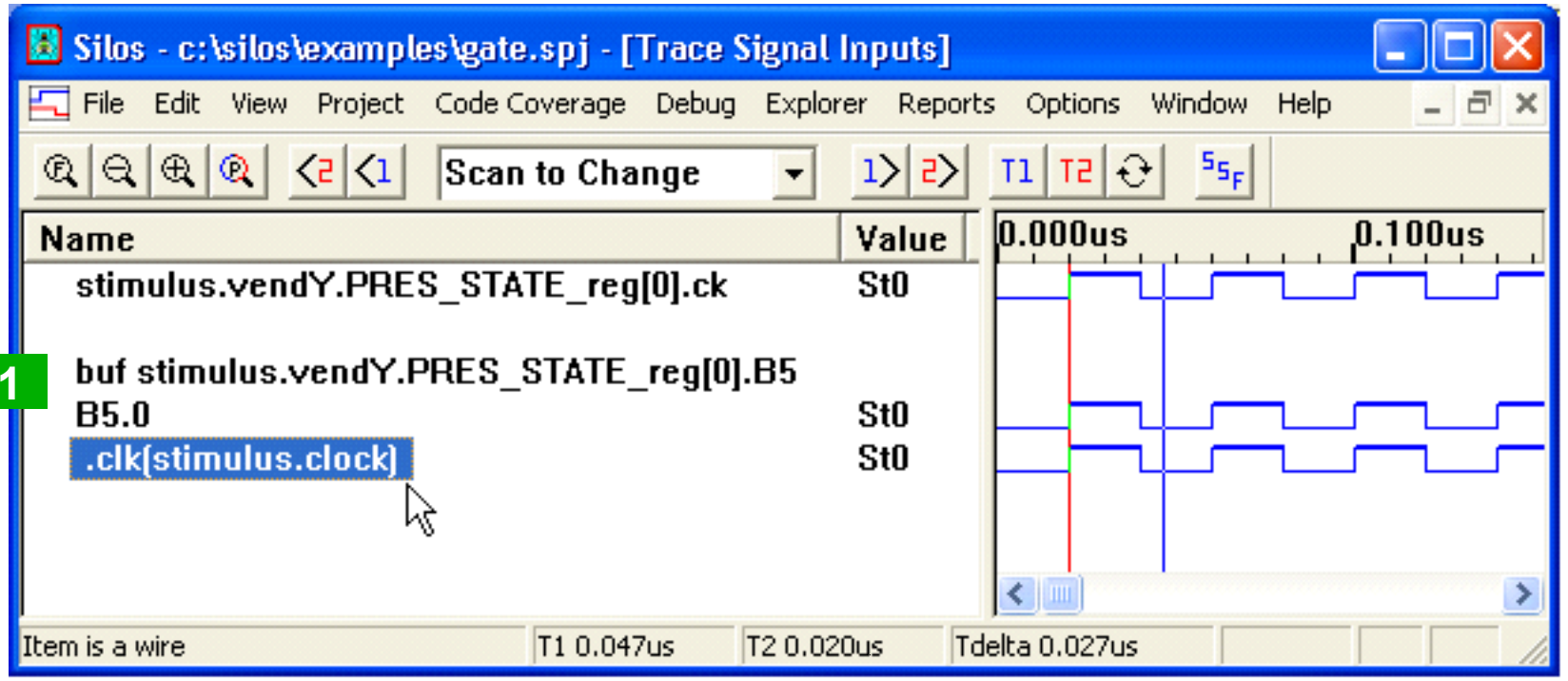


Three input “nand” gate **[1]**.

Double-click on the “ck” **[2]** input to the “nand” gate. The Low to High change on “ck” caused the “nand” gate to change.

SILVACO

Gate Level Debugging

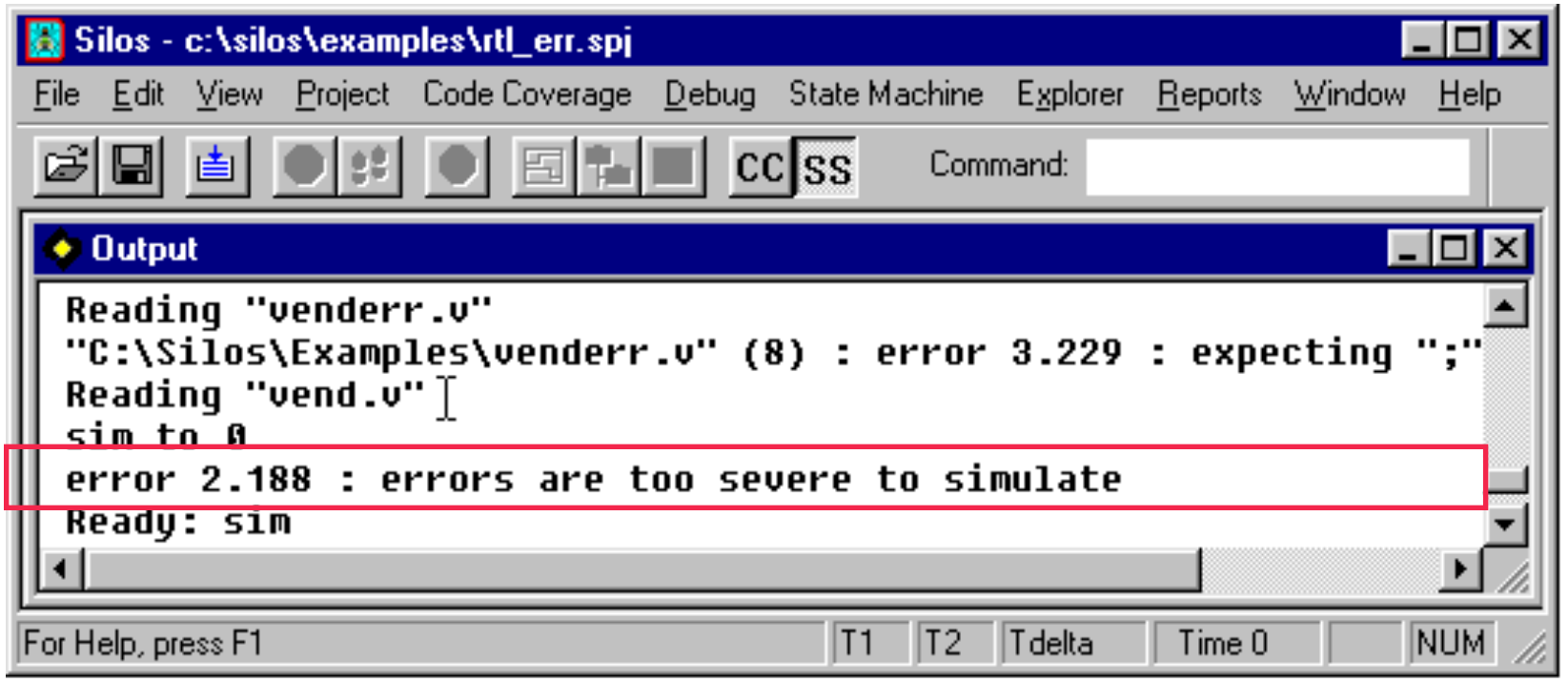


Buffer gate [1]

The original unknown to low transition has been traced back to signal “clock” in the testbench.

SILVACO

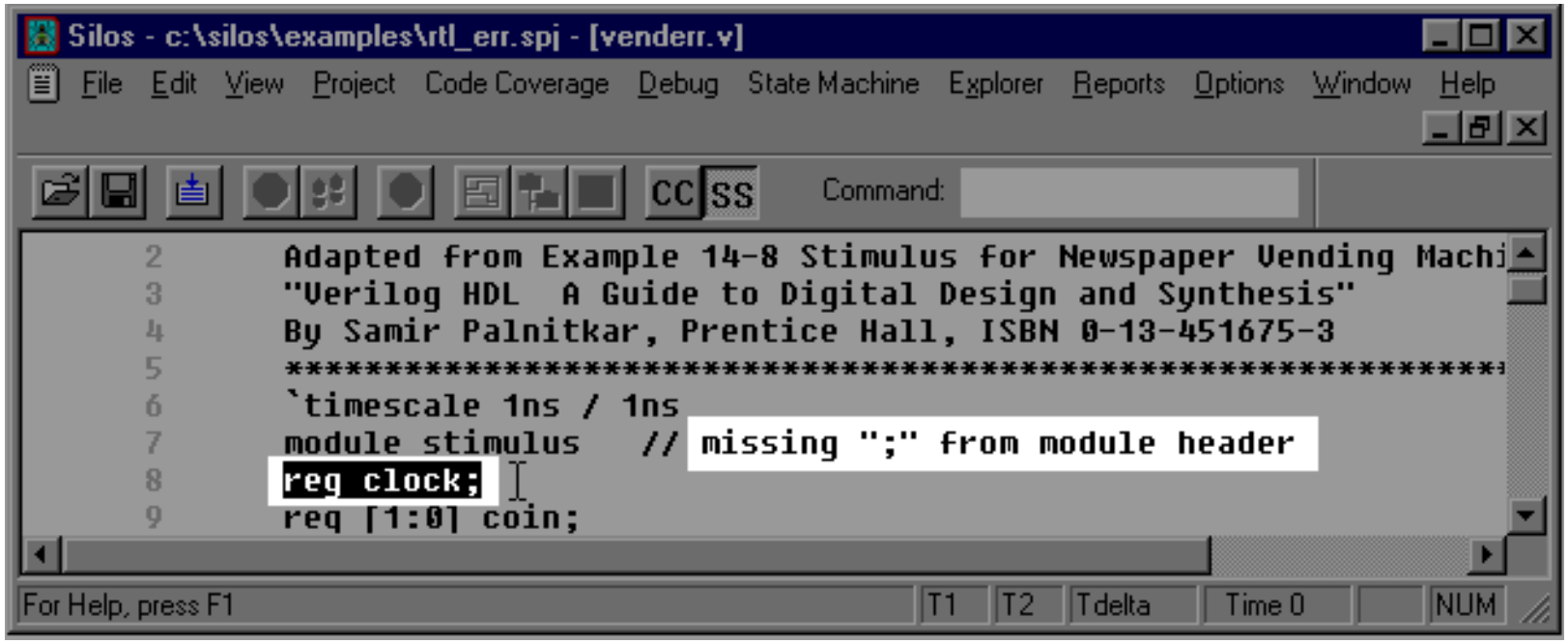
Errors



Double click on the error message to automatically open the source file and see the error.

SILVACO

Errors

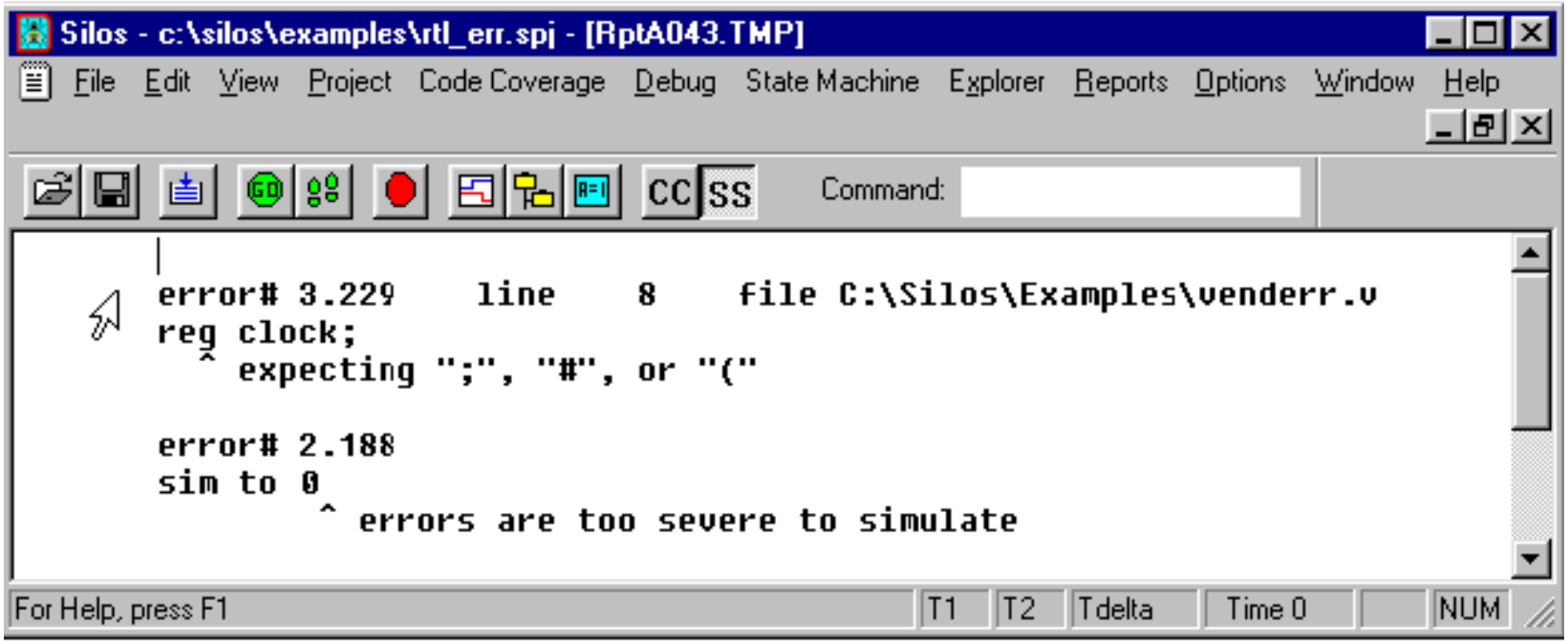

Silos - c:\silos\examples\rtl_err.spj - [venderr.v]
File Edit View Project Code Coverage Debug State Machine Explorer Reports Options Window Help
CC SS
Command:
2 Adapted from Example 14-8 Stimulus for Newspaper Vending Machi
3 "Verilog HDL A Guide to Digital Design and Synthesis"
4 By Samir Palnitkar, Prentice Hall, ISBN 0-13-451675-3
5 **********
6 `timescale 1ns / 1ns
7 module stimulus // missing ";" from module header
8 reg clock;
9 reg [1:0] coin;
For Help, press F1
T1
T2
Tdelta
Time 0
NUM

Silos III highlights the next line because the error could not be detected until that line is interpreted.

Error is caused by semicolon ";" missing from the module header.

SILVACO

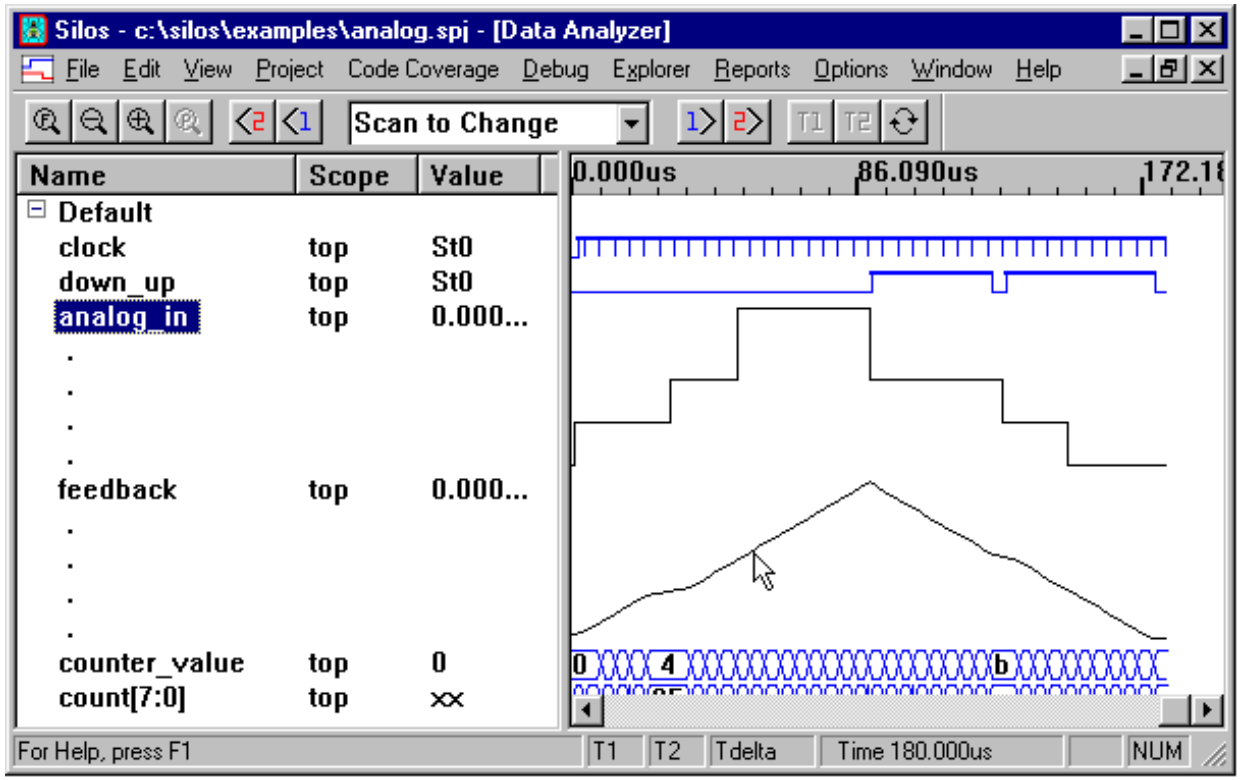
Errors



Error report for Silos III.

SILVACO

Analog Waveforms



Stepping function display of analog signal.

Piece-wise linear display of analog signal.

Double-click to toggle between stepping function and piece-wise linear.

SILVACO

Silos Advanced Features

- Supports Industry-Standard PLI 1.0 Interface
  - Currently implementing PLI 2.0
- Silos supports the SDF language as defined by the IEEE 1497 standard for SDF. Silos also supports a commonly used extension used to specify the SDF file of delay values, the “$sdf_annotate” system task.
  - $sdf_annotate(“file_name”, module_instance);
- Interface for SystemC

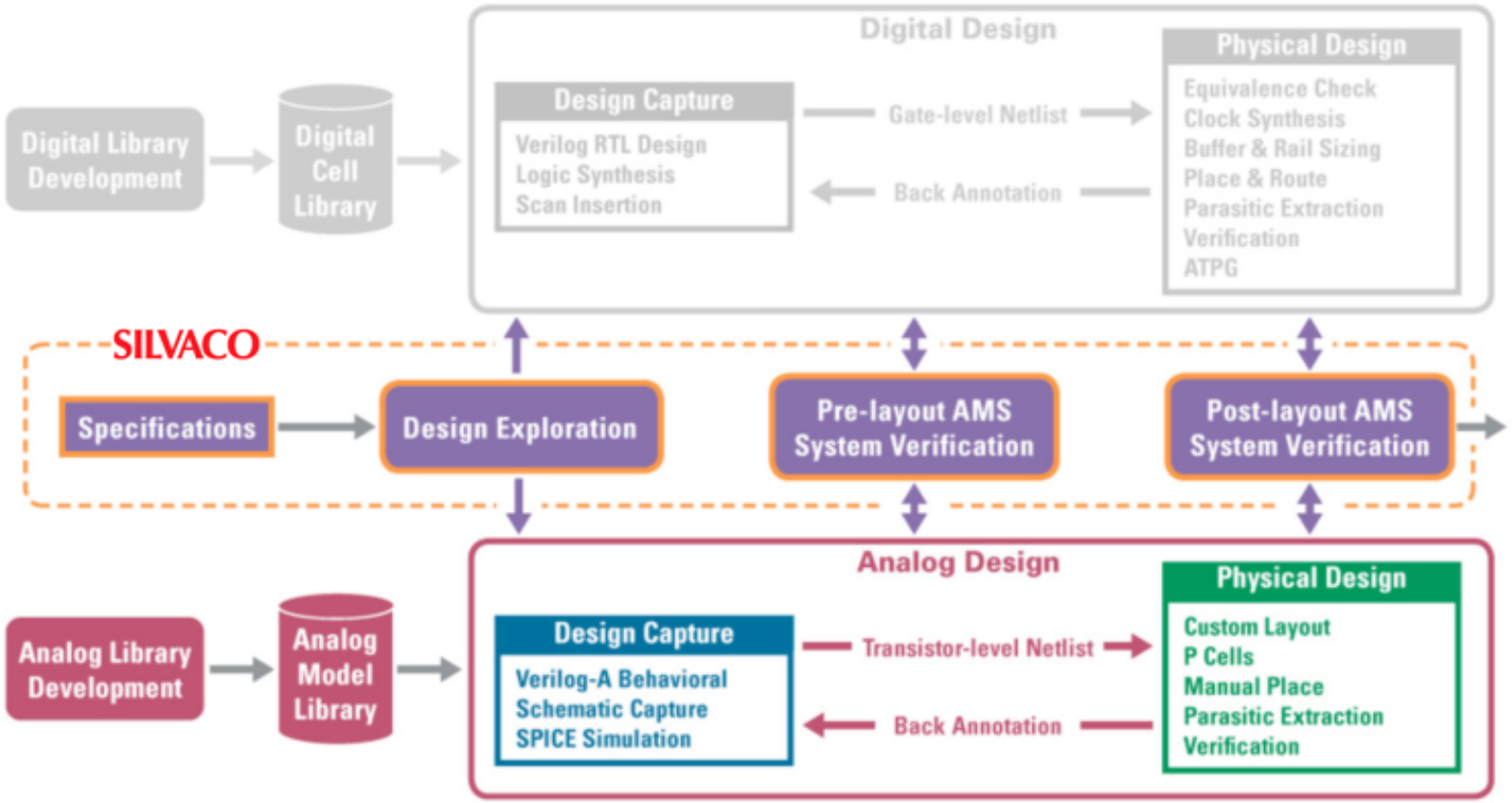
SILVACO

Silos Future – Analog Mixed-Signal Solution

- Silos-AMS beta release in May 2004
  - Integrated power of SmartSpice
  - Solver improvements for simulation speed
  - Common Silos GUI on WINDOWS, SUN, and LINUX platforms
  - STA (Static Timing Analyzer) capability September 2004

SILVACO

Analog and Mixed Signal Design Flow



SILVACO